Machine Learning for Human Action Recognition and Pose Estimation based on 3D Information

Diogo Luvizon

UNIVERSITÉ PARIS SEINE

THÈSE

pour obtenir le grade de docteur délivré par

ÉCOLE DOCTORALE EM2PSI
Économie, Management, Mathématiques, Physique et Sciences Informatiques

Spécialité doctorale "STIC (Sciences et Technologies de l'Information et de la Communication)"

présentée et soutenue publiquement par

Diogo CARBONERA LUVIZON

le 8th Avril 2019

Machine Learning for Human Action Recognition and Pose Estimation based on 3D Information

Directeur de thèse : **David PICARD**
Codirecteur de thèse : **Hedi TABIA**

Jury :

Christian WOLF,	Maître de Conférences, INSA de Lyon et LIRIS	Rapporteur
Élisa FROMONT,	Professeure, Université de Rennes 1	Rapporteure
Marie-Paule CANI,	Professeure, École Polytechnique	Examinatrice
Christian THEOBALT,	Professeur, Max Planck Institute for Informatics	Examinateur
Cordelia SCHMID,	Directeur de Recherche, INRIA	Examinatrice

ETIS
Equipes Traitement de l'Information et Systèmes
UMR CNRS 8051, F-95000 Cergy Pointoise, France

Acknowledgements

I would like to thank my advisers, David Picard and Hedi Tabia, for their dedication and confidence in my work. I will never forget the many hours that we have spend together developing very fruitful ideas. In particular, I would like to express my gratitude to David Picard for his optimism, which was one of the most important factors in keeping me steady.

I also want to thank Dan Vodislav for his support and Nicolas Thome for some interesting discussions about my work. I would specially like to thank Rodrigo Mineto for his initial support, without which this thesis would not be possible. I would like to thank the ETIS laboratory for offering me a well equipped office right next to the coffee machine. I also would like to thank the Brazilian Council for Scientific and Technological Development (CNPq) for the financial support.

I would like to express my eternal gratitude to my parents, Rosa Maria Carbonera and Osvaldo Luvizon, for their enormous effort to provide a good education to me and to my brothers. I would particularly like to thank my beloved wife, Gabriela Luvizon, for her comprehension during my frequent absences and for her constant motivation. Finally, I would like to thank my son, Gael Luvizon, and my daughter, Lina Luvizon, for giving me the opportunity to see the most amazing example of learning system from the beginning.

Context

This work was developed at the ETIS laboratory (UMC 8051, CNRS), attached to the *École National Superieur d'Electronique et ses Applications* (ENSEA) and the Paris Seine University, from October 2015 to March 2019. The research was supported by the Brazilian National Council for Scientific and Technological Development (CNPq), as part of the Brazilian national program "Ciências sem Fronteiras" (science without borders), which one of the main objectives is to promote national outstanding students to perform high quality research in international universities.

Summary

3D human action recognition is a challenging task due to the complexity of human movements and to the variety on poses and actions performed by distinct subjects. Recent technologies based on depth sensors can provide 3D human skeletons with low computational cost, which is an useful information for action recognition. However, such low cost sensors are restricted to controlled environment and frequently output noisy data. Meanwhile, convolutional neural networks (CNN) have shown significant improvements on both action recognition and 3D human pose estimation from RGB images. Despite being closely related problems, the two tasks are frequently handled separated in the literature. In this work, we analyze the problem of 3D human action recognition in two scenarios: first, we explore spatial and temporal features from human skeletons, which are aggregated by a shallow metric learning approach. In the second scenario, we not only show that precise 3D poses are beneficial to action recognition, but also that both tasks can be efficiently performed by a single deep neural network and still achieves state-of-the-art results. Additionally, we demonstrate that optimization from end-to-end using poses as an intermediate constraint leads to significantly higher accuracy on the action task than separated learning. Finally, we propose a new scalable architecture for real-time 3D pose estimation and action recognition simultaneously, which offers a range of performance vs speed trade-off with a single multimodal and multitask training procedure.

Résumé

La reconnaissance d'actions humaines en 3D est une tâche difficile en raison de la complexité de mouvements humains et de la variété des poses et des actions accomplies par différents sujets. Les technologies récentes basées sur des capteurs de profondeur peuvent fournir les représentations squelettiques à faible coût de calcul, ce qui est une information utile pour la reconnaissance d'actions. Cependant, ce type de capteurs se limite à des environnements contrôlés et génère fréquemment des données bruitées. Parallèlement à ces avancées technologiques, les réseaux de neurones convolutifs (CNN) ont montré des améliorations significatives pour la reconnaissance d'actions et pour l'estimation de la pose humaine en 3D à partir des images couleurs. Même si ces problèmes sont étroitement liés, les deux tâches sont souvent traitées séparément dans la littérature. Dans ce travail, nous analysons le problème de la reconnaissance d'actions humaines dans deux scénarios: premièrement, nous explorons les caractéristiques spatiales et temporelles à partir de représentations de squelettes humains, et qui sont agrégées par une méthode d'apprentissage de métrique. Dans le deuxième scénario, nous montrons non seulement l'importance de la précision de la pose en 3D pour la reconnaissance d'actions, mais aussi que les deux tâches peuvent être efficacement effectuées par un seul réseau de neurones profond capable d'obtenir des résultats du niveau de l'état de l'art. De plus, nous démontrons que l'optimisation de bout en bout en utilisant la pose comme contrainte intermédiaire conduit à une précision plus élevée sur la tâche de reconnaissance d'action que l'apprentissage séparé de ces tâches. Enfin, nous proposons une nouvelle architecture adaptable pour l'estimation de la pose en 3D et la reconnaissance de l'actions simultanément et en temps réel. Cette architecture offre une gamme de compromis performances vs vitesse avec une seule procédure d'entraînement multitâche et multimodale.

Publications

International Journals

- D. C. Luvizon, H. Tabia, D. Picard. **Learning features combination for human action recognition from skeleton sequences**. *Pattern Recognition Letters*, volume 99, pages 13-20, 2017.

International Conferences

- D. C. Luvizon, D. Picard, H. Tabia. **2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning**. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5137-5146, 2018.

French Conferences

- D. C. Luvizon, D. Picard, H. Tabia. **Multimodal Deep Neural Networks for Pose Estimation and Action Recognition**, *Congrès Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), oral presentation*, 2018.

Pre-Print

- D. C. Luvizon, H. Tabia, D. Picard. **Human Pose Regression by Combining Indirect Part Detection and Contextual Information**. *CoRR, abs/1710.0232*, 2017.
- D. C. Luvizon, H. Tabia, D. Picard. **SSP-Net: Scalable Sequential Pyramid Networks for Real-Time 3D Human Pose Regression**, 2018.
- D. C. Luvizon, H. Tabia, D. Picard. **Multitask Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition**, 2019.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context and Motivation

As a natural result of evolution, humans are very efficient in recognizing other humans and their behavior. The way that human beings perceive other humans is strongly based on visual observation and interpretation. Consequently, making machines understand human behavior is one of the greatest challenges in computer science. The field of computer science aiming, but not restricted, to enable computer to achieve such a high-level understanding of visual content is called computer vision. Among several different areas of research, two tasks related to computer vision are estimating human poses and recognizing human actions. Nowadays, automating human pose estimation and action recognition has become an essential step towards many other important applications, such as automatic surveillance systems, human-computer interfaces, sports performance analysis, augmented reality, 3D scene understanding, content-based video and image indexation, among many others.

Despite the considerable progress of computer vision algorithms from the last two or three decades, computers are still not effective in solving some tasks which involve complex and high-level semantics. To tackle such problems, machine learning is frequently used to create statistical models from training data manually annotated by humans. As a consequence, building machine learning algorithms capable of solving complex computer vision problems is recently one of the biggest challenges in the domain.

In this thesis, we target the problems of human action recognition and human pose estimation from the machine learning perspective. Both problems are strongly related, not only because understanding the human body is a common key aspect for the two tasks, but also because the human pose information is of great relevance for action recognition. This observation leads to two premises. First, the human action recognition task benefits from the human pose information, and second, since the two problems are strongly related, they could be handled jointly in a better way than separately.

With the recent progress of machine learning and specially deep learning algorithms, complex problems have been successively addressed by end-to-end optimization [53, 46, 47]. We believe

that it is not different for action recognition, and from this emerges our third premise: optimizing one complex task from end-to-end is better than dividing the problem into subtasks with individual optimization.

Based on the three premises stated before, our goal is to handle the high-level problem of human action recognition as a human pose dependent problem. To this end, reliable pose estimation is essential. The human poses or skeletal representation give the spatial coordinates of body parts, which is a highly discriminant information to recognize certain actions dependent on position and movements of the human body, but also provide a relevant information about where to focus to extract visual information, which could be useful to identify action-related objects and interactions between the person performing an action and its environment.

To avoid the complex task of estimating the human pose from monocular images, depth sensors can be used to produce depth maps, which are invariant to color and texture aspects and facilitate the segmentation of 3D space. The human pose can be estimated from depth maps, but the reliability of estimations depends on the pose itself and on the interaction with objects in the scene, since no visual aspects are available. As a result, cases of failure are common, as depicted in Figure 1.1.

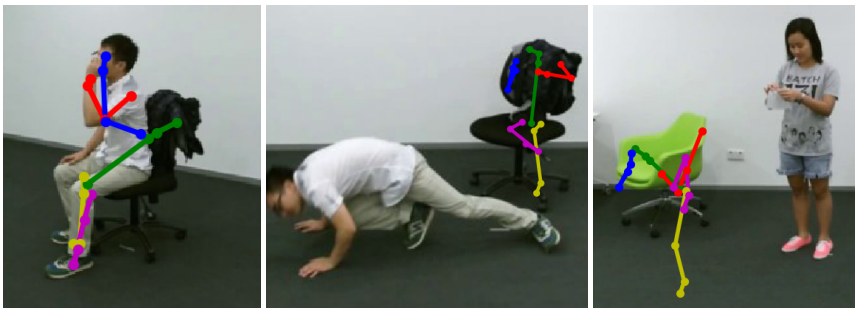


Figure 1.1 – Failure cases of human skeleton estimation from Microsoft's Kinect.

Another possibility is the more difficult task of estimating human poses from monocular images. However, when considering optimization from end-to-end, the current most reliable methods for monocular pose estimation are not fully differentiable, since they cast pose estimation as a per-pixel body-part classification problem by predicting heat maps instead of body joint locations. This prevents such approaches from being used as a building block for action, considering our third premise. This can be possibly one of the reasons that such methods are not frequently extended in the literature to perform action recognition.

Considering the exposed ideas and limitations, we present the key contributions of this thesis in the following section.

1.2 Key Contributions

As our first contribution, we propose a new framework for human action recognition by exploring human skeleton sequences captured by the Kinect sensor. From the skeleton sequences, we propose to extract localized features, considering position and motion, which are then aggregated to form a global feature representation. By using a shallow metric learning approach, we are able to learn a combination of features with the objective to better distinguish between different actions. The promising results show that the skeletal or pose representation is very relevant to recognize some actions, supporting our first premise that action recognition benefits from pose.

Considering the limitations of depth based skeleton estimation and the partial differentiability of current state-of-the-art monocular pose estimation methods, we propose a new human pose regression approach from RGB images that is fully differentiable and achieves state-of-the-art results on 3D human pose estimation and comparable results on 2D scenarios. The proposed method is trainable from end-to-end and predicts human poses in body joints coordinates. As a byproduct, our method also provides body joint probability maps, corresponding to the regions in the image containing the human body parts.

We also investigate different network architectures and 3D pose regression variants, resulting in a new convolutional neural network (CNN) architecture able to perform precise and fast human pose inferences. Additionally, we investigate the problem of predicting human poses in absolute coordinates, which could be specially useful when performing predictions with multiple cameras.

Based on our differentiable approach for human pose estimation from RGB images, we build on top of it a human action recognition method, considering our three premises: First, our method is based on reliable and robust 3D human pose estimation. Second, the two tasks, pose estimation and action recognition, can be performed simultaneously in a multitask fashion; and third, the multitask method can be optimized from end-to-end. Finally, we demonstrate by our method and by our results each of our premises.

1.3 Structure of the Thesis

This thesis is divided in six chapters. In chapter 2, we present the bibliographic review, considering the recent methods most related to our work. In chapter 3, we propose a new framework for human action recognition from skeleton sequences obtained from depth sensors. This chapter is based on the following publication:

- D. C. Luvizon, H. Tabia, D. Picard. **Learning features combination for human action recognition from skeleton sequences**. *Pattern Recognition Letters*, volume 99, pages 13-20, 2017.

The proposed approach for human pose regression from RGB images, its extensions to 3D scenarios, and the proposed network architecture for 3D pose estimation are presented in chapter 4, which is based on the following articles:

- D. C. Luvizon, H. Tabia, D. Picard. **Human Pose Regression by Combining Indirect Part Detection and Contextual Information**. *CoRR, abs/1710.0232*, **pre-print**, 2017.
- D. C. Luvizon, D. Picard, H. Tabia. **2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning**. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5137-5146, 2018.
- D. C. Luvizon, H. Tabia, D. Picard. **SSP-Net: Scalable Sequential Pyramid Networks for Real-Time 3D Human Pose Regression**. **Submitted to** *Pattern Recognition*, November 2018.

In chapter 5, we present a fully differentiable, multitask approach, for human action recognition based on predicted poses, considering two scenarios: separated learning and joint multitask optimization. This part is partially based on the previous CVPR'18 paper, in addition to the following articles:

- D. C. Luvizon, D. Picard, H. Tabia. **Multimodal Deep Neural Networks for Pose Estimation and Action Recognition**, *Congrès Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, 2018.

- D. C. Luvizon, D. Picard, H. Tabia. **Multitask Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition**. **Submitted to** *TPAMI*, January 2019.

Finally, in chapter 6, we conclude this thesis and give ours perspectives for future researches.

Chapter 2

Related Work

Contents

Prologue

Context:

In this chapter, we contextualize our work by presenting a non exhaustive review of the related work. Recent methods related to human pose estimation are presented in section 2.1, and previous works on action recognition are discussed in section 2.2. We also discuss the limitations of current approaches and highlight our contributions to each domain.

In the literature, both terms “human skeleton” and “human pose” are used to designate a high level representation of the human body. Frequently, “human skeleton” is used to refer to a 3D human body representation estimated from depth maps by using depth sensors, such as the Microsoft’s Kinect. On the other hand, “human pose” is a more generic term used for both 2D manually annotated keypoints (or landmarks), as well as for 3D representations computed by motion capture (MoCap) systems. In order to clarify the different sources of information in this work, we use these two terms accordingly to this frequently used standard from previous work.

2.1 Human Skeleton and Human Pose Estimation

In this section, we review some of the methods for human skeleton and human pose estimation most related to our work, which we divide into five sections:

- *Human Skeleton Prediction using Depth Sensors*, we present some techniques and limitations about human skeleton estimation from depth maps.
- *2D Human Pose Estimation from RGB Images.* Here we present some methods for 2D pose estimation from RGB images using classical and deep architectures.
- *Monocular 3D Human Pose Estimation*, we present the state-of-the-art methods for 3D pose prediction and their limitations.
- *Multi-stage Architectures for Human Pose Estimation*, we detail some aspects of current multi-stage CNN architectures for pose estimation.
- *Multi-person Pose Estimation*, we present the few methods targeting 3D multi-person pose estimation.

For a complete and detailed bibliographic review, we encourage the readers to refer to the 3D human pose estimation survey [121].

2.1.1 Human Skeleton Prediction using Depth Sensors

In the last decade, the rising of consumer depth sensors such as the Microsoft's Kinect [91] and the Asus' Xtion [5] have benefited many applications from depth and color data [45, 169]. These sensors are based on an infrared (IR) projector, an IR camera, and a color camera, resulting in a capturing system able to record both RGB and depth data (RGB-D). Among different applications in computer vision, depth sensors motivated the development of algorithms which estimate the human skeleton in real-time from depth maps [126, 41, 125, 150, 6], specially due to the invariance of depth maps to color and texture aspects. For example, as illustrated in Figure 2.1, Shotton et al. [125] proposed an algorithm based on deep randomized decision forest to discriminate body parts, which are then used to predict 3D body joint locations. The algorithm runs at real-time on the Xbox 360 GPU and is used as the standard tool for skeleton estimation from Kinect data.

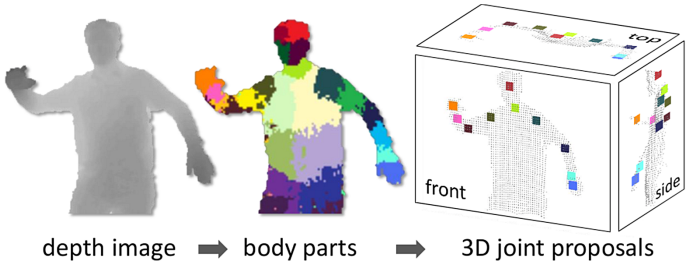


Figure 2.1 – Human skeleton estimation from depth maps using detected body parts. Adapted from Shotton et al. [125].

Despite providing additional information, depth sensors have some disadvantages compared to standard monocular cameras:

- Depth sensors are limited to controlled and indoor environment, since they rely on IR sensing.

- Compared to RGB images, RGB-D frames frequently have lower resolution, since depth sensors are more computationally expensive.
- In cost-effective depth sensors, the color and depth channels are frequently not synchronized and/or do not have a pixel-wise correspondence, which hinders the optimal use of both channels together.
- RGB videos are widely available on the Internet (e.g., YouTube and Flickr), while RGB-D videos are mostly limited to a few academic datasets.

The above mentioned limitations and the breakthrough of deep learning and convolutional neural networks motivated us to study human pose estimation from traditional color images, as detailed in the following sections.

2.1.2 2D Human Pose Estimation from RGB Images

The problem of 2D human pose estimation from RGB images has been intensively studied during the last 10 years, from Pictorial Structures [4, 31, 103] to more recent CNN approaches [95, 77, 105, 54, 111, 149, 10, 137, 139, 102, 101]. From the literature, we can distinguish two families of methods for pose estimation: detection based and regression based methods. The former family of methods tries to detect body joints separately, which are further combined, resulting in the final predicted pose from aggregated parts. In the latter, the methods are able to map directly input images to body joints coordinates, usually by using a non-linear regression function. We provide some examples from each family of methods in the two following sections.

2.1.2.1 Detection based Approaches

Pischulin et al. [105] proposed DeepCut, a graph cutting algorithm that relies on body parts detected by DeepPose. This method has been improved in [54] by replacing the previous CNN by a deep Residual Network (ResNet) [48], resulting in very competitive accuracy results, specially on multi-person detection.

More recent 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 [111, 10, 13, 93, 29]. Specifically, such methods employ a deep CNN to predict one heat map per body joint, which are learned to reproduce Gaussian ground truth heat maps, as illustrated in Figure 2.2. Based on this approach, Bulat et al. [13] proposed a two-stages CNN for coarse and fine heat

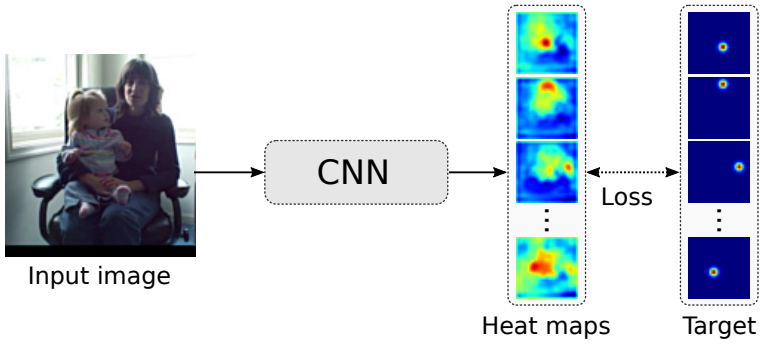


Figure 2.2 – Overview of recent detection based approaches for heat maps regression and pose estimation.

map regression using pre-trained models. Gkioxari et al. [42] presented a structured prediction

method, where the prediction of each joint depends on the intermediate feature maps and the distribution probability of the previously predicted joints.

Following the tendency of deeper models with residual connections, Newell et al. [93] proposed the Stacked Hourglass (SH) networks with convolutions in multi-level features, allowing reevaluation of previous estimations due to a stacked block architecture with many intermediate supervisions. The part-based learning process can benefit from intermediate supervision because it acts as constraints on the lower level layers. As a result, the feature maps on higher levels tend to be cleaner.

Since the release of the SH networks, methods in the state of the art are proposing complex variations of this architecture. For example, Chu et al. [29] proposed an attention model based on Conditional Random Field (CRF) and Yang et al. [155] replaced the residual unit from [93] by the Pyramid Residual Module (PRM). With the emergence of Generative Adversarial Networks (GANs) [43], Chou et al. [27] 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 [23].

All the previous methods that are based on detection need additional steps on training to produce artificial ground truth from joint positions, which represent an additional processing stage and additional hyper parameters, since the ground truth heat maps have to be defined by hand, usually as a 2D Gaussian distribution centered on ground truth locations. On evaluation, the inverse operation is required, i.e., heat maps have to be converted to joint positions, generally using the argument of the maximum a posteriori probability (MAP), called *argmax*. Consequently, in order to achieve good precision, predicted heat maps need reasonable spatial resolution (i.e., number of pixels encoded in one activation), which increases quadratically the computational cost and memory usage. On the other hand, as detailed in the following section, regression based approaches output poses in (x, y) for 2D or (x, y, z) for 3D coordinates, preventing from requiring discretization, artificially generated ground truth, and post-processing stages.

2.1.2.2 Regression based Approaches

Some methods tackle pose estimation as a keypoint regression problem, where a nonlinear function is used to map input images directly to joint coordinates. One of the first regression approaches was proposed by Toshev and Szegedy [139] as a holistic solution based on cascade regression for body part detection, where individual joint positions are recursively improved, taking a full frame as input. Pfister et al. [102] proposed the Temporal Pose ConvNet to track upper body parts, and Carreira et al. [16] proposed the Iterative Error Feedback by injecting the prediction error back to the input space, improving estimations recursively. More recently, Sun et al. [130] proposed a structured bone based representation for human pose, which is statistically less variant than absolute joint positions and can be indistinctly used for both 2D and 3D representations. However, the method requires converting pose data to the relative bone based format. Moreover, those results are all outperformed by detection based methods, mainly because traditional regression approaches are sub-optimized for the highly complex task of pose estimation. This is an evidence that regressing coordinates is a difficult problem.

In order to tackle this weakness, we propose a different strategy by replacing the argmax from recent part-based methods by the *soft-argmax*, allowing sub-pixel accuracy while being fully differentiable. The main advantage of a differentiable method is that the output of the pose estima-

tion can be used in further processing and the whole system can be fine-tuned.

2.1.3 Monocular 3D Human Pose Estimation

Estimating the human body joints in 3D coordinates from monocular RGB images is a very challenging problem with a vast bibliography available in the literature [56, 107, 38, 134, 74, 73, 55, 172]. Despite the majority of 2D pose estimation methods being detection based approaches [93, 52], 3D pose estimation is mostly handled as a regression problem [83, 88, 1, 171, 131, 94]. One of the reasons is due to the additional third dimension, which significantly increases the complexity of classification based solutions.

A common approach for 3D human pose estimation is to lift 3D predictions from 2D poses estimated with keypoint detectors [108, 70, 114, 158]. For example, Martinez et al. [87] proposed a baseline architecture for learning 3D representations from 2D poses. Chen and Ramanan [19] proposed a two stage method: first 2D poses are estimated in the camera space, then the predicted 2D points are used to match a non parametric shape model, from which 3D predictions are obtained. Structural constraints have also been exploited to penalize invalid angles and segments [30], resulting in more realistic predictions. Despite being more robust to visual variations, lifting 3D poses from 2D points is an ill-defined problem, which frequently results in ambiguity.

Deep CNN architectures have been used to learn precise 3D representations from RGB images [173, 136, 87, 135, 88] thanks to the availability of high precise 3D annotated data [57]. Many methods are now surpassing methods which use depth-sensors [90]. Mehta et al. [88] proposed an improved supervision scheme with one auxiliary task for 2D heat maps prediction and a second order kinematic relation, resulting in a multimodal approach for normalized 3D human pose estimation. This method was extended to VNect [90], a real-time 3D human pose estimation system from monocular images, by replacing the fully-connected layer for 3D pose by location maps that encode the relative (x, y, z) coordinates for each joint. Additionally, the person's bounding box is tracked based on the 2D predictions, and a temporal filtering is applied to stabilize predictions. In their method, the authors assume that all persons have the same scale and the same height, so given the camera calibration, the estimated 2D pose (in image coordinates), and the relative 3D pose centered on the root joint, the global pose can be estimated in the camera coordinate system.

Pavlakos et al. [99] proposed the volumetric stacked hourglass architecture, but the method suffers from the significant increase in the number of parameters and in the required memory to store all the gradients. More recently, Yang et al. [158] proposed to use an adversarial network to distinguish between generated and ground truth poses, resulting in improvements on predictions on uncontrolled environments. Compared to [99], we show in our work that (i) smaller volumetric heat maps can be used with soft-argmax and still improve results, since soft-argmax is a continuous regression function, and (ii), the volumetric representation can be fully replaced by predicting 2D depth maps, that encode the depth related to each body joint, resulting in even lower computational complexity and better results. Sun et al. [132] proposed a similar approach to the soft-argmax, called *integral regression*. Their work was developed concurrently and independently form ours. However, the method proposed in [132] still depends on artificial heat maps generation for intermediate supervision and on a costly voxilized representation for 3D predictions.

Another challenge related to 3D pose estimation is the lack of rich visual data. Since precise 3D annotation depends on expensive and complex Motion Capture (MoCap) systems, public datasets are usually collected in controlled environments with static and clean background, despite having few subjects. To alleviate this limitation, Mehta et al. [88] proposed to first train a 2D model on

data collected "in-the-wild" with manual 2D annotations, and then to use transfer learning to build a 3D network that predicts 3D joint positions. More recently, synthetic data augmentation have been proposed to alleviate the lack of "in-the-wild" images with 3D pose annotations [116], e.g., the SURREAL dataset [142] and the MuPoTS-3D dataset [89]. However, such solutions are still not optimal, since it is very difficult to place an avatar over natural images while keeping the context realistic, as can be observed in Figure 2.3.

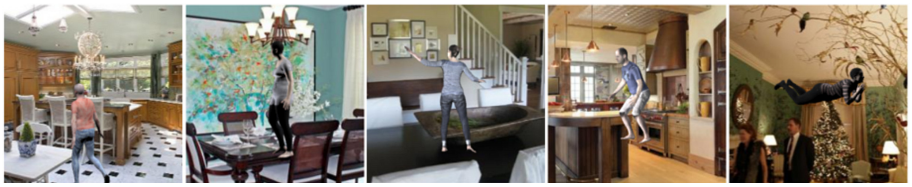


Figure 2.3 – Samples of synthetic data from the SURREAL dataset. Adapted from Varol et al. [142].

Differently, in our approach we merge images from 3D datasets with precise annotations but still background and images "in-the-wild" with manually 2D annotations in a single training procedure. This is possible because in our method we do not backpropagate the error relative to the z coordinate for 2D input data. Additionally, we propose 3D structural constraints on predicted poses, which can be applied to any input image, independently on the available labels.

2.1.4 Multi-stage Architectures for Human Pose Estimation

Multi-stage architectures have been widely used for human pose estimation, specially for the more established problem of 2D pose [10, 42, 29, 149]. The main reason for the success of such architectures is the successive improvements provided by intermediate supervision. A common practice in previous methods is to regress heat maps representations, which correspond to a score map for a given keypoint being present at a given location. The refinement of such heat maps is crucial for achieving good precision, as noted in [13] and extended in the SH network [93], where a sequence of U-nets with multi-scale processing is used to refine a set of predicted heat maps. On 3D scenarios, Zhou et al. [170] benefits from 2D heat maps to guide a 3D pose regressor, introducing a weakly-supervised approach for lifting 3D predictions from 2D data.

As observed in [93], intermediate supervisions are beneficial for good precision, since they allow the network to refine predictions successively. However, traditional detection based methods cannot easily benefit from multi-scale intermediate supervision, since such methods are not scale invariant. On the other hand, the proposed regression approach is invariant to the scale of feature maps (as shown in section 4.5), allowing multi-scale supervision in a straightforward way.

2.1.5 Multi-person Pose Estimation

Several methods for 2D multi-person pose estimation have been proposed in the literature [52, 105, 54, 92, 15, 60, 71]. A simple solution is to first localize the persons, then estimate the pose on each localisation [106, 58, 98]. However, these methods not only depend on the robustness of the person detector, but also introduce redundant computations. A more efficient approach is to detect body keypoints with a CNN classifier [105, 54]. Even though such methods require a post-processing stage, it is often computationally less expensive than feeding multiple bounding boxes to a single pose estimator.

To the best of our knowledge, only three methods have been proposed to tackle multi-person 3D pose estimation [117, 165, 89]. The pioneer work is the LCR-Net [117] architecture, on which

the classification output from an R-CNN [115] is replaced by a pose classification task. Pose proposals are then fed to a regression module for further refinement. However, the method is limited to predictions relative to the root joint (no absolute z for each person) and by the number of predefined anchor-poses, resulting in lower precision on single person when compared to regression methods [131]. Zanfir et al. [165] proposed to reconstruct 3D pose and shape by integrating physical scene constrains and temporal coherence, and Mehta et al. [89] proposed a regression method based on the Occlusion-Robuts Pose-Maps (ORPMs).

One of the reasons for the scarcity of works on multi-person 3D pose estimation is the lack of abundant and natural 3D data with multiple person (except for synthetic data as in [89]), which is a requirement for many recent data-driven approaches.

2.2 Human Action Recognition

In this section, we present some of the methods for human action recognition most related to our work, which are structured as:

- *Action Recognition from Skeleton Sequences and Depth Maps*, on which we focus on action recognition methods using depth sensors.
- *Action Recognition from RGB Image Sequences*. In this part we present the methods related to our work that use RGB image sequences as input.

For a general literature review on the subject, considering both skeleton and RGB based methods, we encourage the readers to refer to the surveys [49, 110, 109].

2.2.1 Action Recognition from Skeleton Sequences and Depth Maps

Recent methods for human action recognition using depth sensors are mostly based on depth maps, skeleton joint sequences, or both [76, 100]. Skeleton sequences are frequently used for action recognition because they encode high level information, both on spatial and temporal domains, with very few feature values, despite being invariant to the visual aspects of the subject. Depth sensors have also some advantages over color cameras, like their invariance to lightning and color conditions and their capability to provide a 3D structure of the scene, which makes the segmentation step easier. Some methods use solely depth maps for action recognition [75, 81, 96, 112, 159]. However, these methods suffer from noisy depth maps and occlusions. To deal with multichannel RGB-D frames, Tran and Ly [140] proposed a latter feature combination scheme, although it can be costly if several features are used, since their method needs one classifier and one weight coefficient per features individually. Using both depth maps and skeleton joints, Wang et al. [148] proposed the actionlet ensemble model. The actionlet features combine the relative 3D position of subgroups of skeleton joints and the local occupancy pattern (LOP) descriptor. To capture the temporal structure of actions, the authors employ the short time Fourier transform on concatenated features to compose a final feature vector.

Based only on skeletons extracted from depth maps, Xia et al. [153] proposed a representation of human pose by the histogram of 3D joints (HOJ3D). They project the sequence of histograms using LDA and label each posture using the k-means algorithm. Each posture label from a sequence is fed into a discrete hidden Markov model (HMM) that gives the matching probability for each action. Their approach showed low accuracy in cross subject tests due to the high intra-class variance observed in the evaluated datasets. In order to reduce the effect of intra-class variance,

several techniques have been applied on feature space, such as sparse dictionary learning [82] and metric learning on Riemannian [33] and Grassmann [128] manifolds. Other methods try to explicitly model the skeleton sequences accordingly to their 3D geometric constraints [144] or by applying specialized graph-based models on the skeleton joints structure [72], which try to explore the geometric relations between joints and to detect more relevant patterns related to actions. Finally, Zanfir et al. [166] showed that speed and acceleration of skeleton joints are also important for action classification by proposing the non parametric moving pose (MP) descriptor. In our first approach for action recognition based on skeleton data (chapter 3), we try to extract relevant pose features by combining joints into subgroups, and by explicitly exploring body movement (speed) from displacement vectors.

With the success of Recurrent Neural Networks (RNNs) in the text and speech domains, some RNN based methods for action recognition from skeleton sequences were also proposed [143, 174]. However, the low range and low resolution of cost-effective depth sensor frequently result in noisy skeletons, which, combined with the limited size of some datasets, makes the learning task difficult. To cope with this noisy data, Spatio-Temporal LSTM networks have been proposed by applying a gating mechanism [78] to learn the reliability of skeleton sequences or by using attention mechanisms [79, 129]. In addition to the skeleton data, multimodal approaches can also benefit from visual cues [124]. In that direction, Baradel et al. [8] 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 image patches around the hand regions. Although the majority of recent methods based on deep neural networks employ LSTM units to model the temporal aspect of actions, they are limited to 20 frames when reporting experiments. This short amount of samples can be easily encoded by sufficiently deep convolutional networks. Since our deep 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.

From the above mentioned works we can conclude two important facts. *First*, both spatial and temporal information are relevant for action recognition, as well as visual cues. However, since they have different nature, it is not trivial to combine them. *Second*, certain joints are more discriminant for specific actions. In the first part of this thesis, we demonstrate that spatial and temporal features can be used together if they are combined correctly [85]. Furthermore, the proposed approach based on skeleton data relies on an aggregation method that preserves fundamental information from individual joints, allowing further efficient metric learning. In the proposed deep architecture, we are able to combine temporal, spatial and visual information in a seamless way, making the best of the input data.

2.2.2 Action Recognition from RGB Image Sequences

Action recognition from videos or RGB image sequences is considered a difficult problem because it involves high level abstraction, high intra-class variations, background clutter, and the temporal dimension is not easily handled. However, contrarily to depth maps, color images are widely available, are not restricted to controlled environments, and provide rich visual information, which is of great relevance to recognize actions contextualized by human-object interactions.

A common approach in the literature to address the task of action recognition is by exploring

localized features in space and time [154, 63, 69, 64, 34]. A good technique to aggregate motion and visual features is by conditioning both information using the human pose to guide features extraction, as demonstrated by [24]. The local motion information is also of great relevance for action recognition, classically described as Dense Trajectories (DT) [146] and Improved Dense Trajectories (IDT) [147]. In all those approaches, the long-term temporal information is usually not taken into account, since localized features are aggregated by a bag-of-features encoding technique for final action classification.

With the advent of deep convolutional neural networks, the temporal dimension can be better incorporated by means of temporal convolutions, which in practice are 3D convolutions in space and time. Such techniques have recently been stated as the option that gives the highest classification scores [14, 17, 7] in large scale datasets, specially when taking into account long-term dependencies [141]. However, 3D convolutions involve a high number of parameters, which require an elevated amount of memory for training, and cannot efficiently benefit from the abundant still images for training, since the input necessarily has to be a video clip. Another option to integrate the temporal aspect is by analysing motion, usually using the optical flow computed from the frame sequence [24, 35]. Unconstrained temporal and spatial analysis is also a promising approach to tackle action recognition, since it is very likely that, in a sequence of several frames, some very specific regions in a few frames are much more relevant than the remaining parts. Inspired on this observation, Baradel et al. [9] proposed an attention model called Glipse Clouds, which learns to focus on relevant image patches in space and time, aggregating the patterns and soft-assigning each feature to workers that contribute to the final action decision. Multi-view videos can also provide additional information, specially in the cases of occlusion, as demonstrated by [145].

We can notice that many 2D action recognition methods use localized information in space and time to perform action decision. However, the human body joints, when available, are used only as an attention mechanism for features extraction, and not for motion analysis. Moreover, the few methods that directly explore the body joints do not generate it, therefore they are limited to datasets that provide skeletal data. We proposed an approach that alleviates these limitations by performing pose estimation simultaneously with action recognition, exploiting the best capabilities of human poses, to extract visual features and to recover the relative motion information. As a result, our model only needs the input RGB frames while still performing discriminative visual recognition guided by estimated body joints.

2.3 Conclusion

In this chapter, we presented some of the state-of-the-art methods which are most related to our work. We divided these methods into two main groups: human pose estimation and action recognition. For both groups, we discussed some characteristics with respect to the source of information i.e., depth sensors or monocular cameras, and the considered scenario as 2D or 3D spaces.

From previous methods for action recognition based on depth maps and skeleton sequences we can observe that a few subsets of joints are more relevant to specific actions, and that both body position and motion information are important to the final action decision. Moreover, current methods have shown difficulties to handle intra-class variations among similar pairs of actions, specially on small datasets composed of few samples per class, and several methods proposing geometrically constrained manifolds for a better data representation. We tackle these limitations by proposing a shallow framework based on skeleton data only, which differs from the previous work by a robust local features aggregation scheme, which combines small groups of joints for

both position and motion analysis, allowing efficient metric learning intended to select the best features combination for action recognition.

Nonetheless, depth sensors are very limited in terms of conditions of use and availability, making human skeleton prediction from depth maps prohibitive under certain conditions. Alternatively, recent advances in deep learning and on convolutional neural networks have allowed precise human pose estimation from color images. However, the most accurate methods are based on detection and are not differentiable. Therefore, they cannot be easily integrated to build on other tasks such as action recognition in a fully differentiable way. We alleviate this limitation by proposing a new human pose regression approach that has several differences compared to previous works. First, our method departs from requiring artificial heat maps generation for training, non-differentiable quantization for evaluation, and volumetric representations for 3D by predicting pairs of heat maps and depth maps, which are then directly transformed to joint coordinates by the proposed regression approach. This solution allows our method to be trained simultaneously with 2D and 3D annotated data, resulting in more robust predictions. Second, differently from previous architectures, our method has intermediate supervisions at different scales, providing different levels of semantic and resolution, which are all aggregated for better predictions refinement. Third, after a single training procedure, our scalable network can be cut at different positions, providing a vast trade-off for precision vs speed. Finally, the proposed approach is also capable of predicting 3D poses in absolute world coordinates with any assumption about persons scale. All these characteristics allow our method to provide high precise 3D pose estimations in a fully differentiable way, using only RGB frames as input.

Despite providing discriminant information for action recognition, the human poses sequence is frequently not enough to distinguish between actions involving similar movements but different visual context, such as “reading” and “playing with tablet”. From the previous work, we can notice the importance of localized visual features extraction, which are essential to identify the context. However, previous action recognition methods based on color images are not able to extract or does not exploit the human pose information efficiently. We show in our approach that a single model can be used to (i) extract the human pose information and to (ii) recognize human actions, using the human pose both as a high discriminant information and as a prior for deep visual features extraction. We also show that optimization from end-to-end for pose estimation and action recognition leverages accuracy on both tasks, since they are related tasks and can benefit from each other. Finally, contrarily to all previous methods, we are able to train a single model using multimodal data such as single frame 2D “in-the-wild”, highly precise 3D poses, and video clips for action, simultaneously and with multitask optimization, while reaching an efficient trade-off for precision vs speed with a single training procedure.

Chapter 3

Human Action Recognition from Skeleton Sequences

Contents

Prologue

Related Article:

- D. C. Luvizon, H. Tabia, D. Picard. **Learning features combination for human action recognition from skeleton sequences**. *Pattern Recognition Letters*, volume 99, pages 13-20, 2017.

Context:

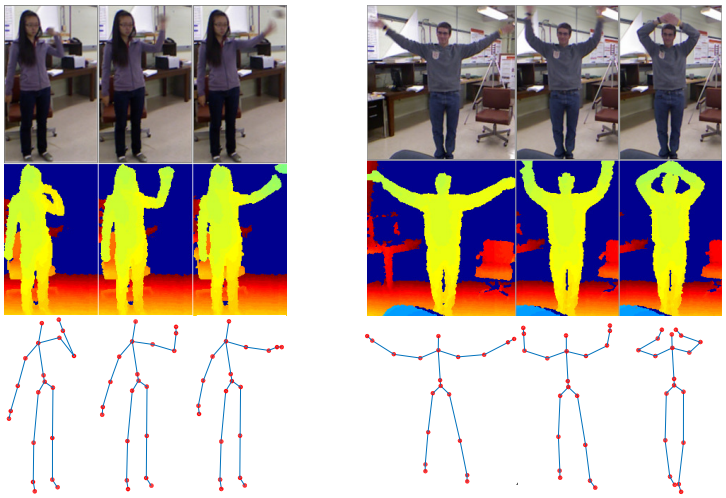
Human action recognition is a challenging task due to the complexity of human movements and to the variety among the same actions performed by distinct subjects. Recent technologies provide the skeletal representation of human body extracted from depth maps, which is a high discriminant information for efficient action recognition. In this chapter, we present a new method for human action recognition from skeleton sequences extracted from RGB-D images. We propose extracting sets of spatial and temporal local features from subgroups of joints, which are aggregated by a robust method based on the Vector of Locally Aggregated Descriptor (VLAD) algorithm and a pool of clusters. Several feature vectors are then combined by a metric learning method inspired by the Large Margin Nearest Neighbor (LMNN) algorithm with the objective to improve the classification accuracy using the nonparametric k-NN classifier. We evaluated our method on

three public datasets, including the MSR-Action3D, the UTKinect-Action3D, and the Florence 3D Actions dataset. As a result, the proposed framework performance overcomes the methods in the state of the art on all the experiments. Additionally, we provide valuable insights about the key aspects of the proposed method, based on our experimental evaluation.

3.1 Introduction

Despite many efforts in the last years, automatic recognition of human actions is still a challenging task. Action recognition is broadly related to human behavior and to machine learning techniques, whereas its applications include improved human-computer interfaces, human-robot interaction and surveillance systems, among others. Activities involving temporal interactions between humans and objects could be handled by graphical models as described in [40]. In this work, we address specifically human actions, which are described as a well defined sequence of movements. Moreover, action recognition methods may be used as intermediate stages in systems capable of providing more complex interpretations such as human behaviour analysis and task recognition [86]. In action recognition frameworks, we can identify three major parts: action segmentation from video streams, modeling and representation of spatial and temporal structure of actions, and action classification. The first two parts are highly dependent on the quality of sensory data, while the classification stage has been proved difficult due to the variety and complexity of human body movements.

Many approaches for human action recognition are based on 2D video streams [152]. However, 2D color images are hard to be segmented and lack depth information. As an alternative to color cameras, depth sensors have been popularized by their low-cost and accessibility. Examples of affordable depth sensors are Microsoft's Kinect and Asus' Xtion, which allows to capture both RGB images and depth maps. The human poses, composed of skeleton joints, can be extracted from depth maps in real-time [125]. Skeleton joints are a high discriminant representation that allows efficient extraction of relevant information for action classification. Samples of RGB images with associated depth maps and skeleton joints from captured human actions are shown in Figure 3.1.



(a) (b)

Figure 3.1 – Samples of RGB images, depth maps, and their respective skeleton joints from the public UTKinect-Action3D dataset [153]. The images correspond to the actions *throw* (a) and *wave hands* (b).

As presented in chapter 2, human action recognition methods are still including some drawbacks, specially when representing the structure of actions. Many authors have proposed to extract spatial features from skeleton joints [144, 153], while others extract temporal information from sequences alignment [82] or by frequency analysis [148] of spatial features. It is known that both spatial and temporal information are fundamental for action recognition, however, an early combination of distinct features may not effectively improve results [140]. Some authors have no-

ticed that the relevance of each skeleton joint varies from one action to another. Wang et al. [148] empirically demonstrated the benefit of grouping certain joints into subgroups to construct more discriminant features. Also, human actions are often performed in very different durations, which demands a robust method to represent the variety of lengths of the input sequences.

Considering the introduced difficulties, we present the three main contributions of this chapter as follows. *First*, we bring traditional methods from the image classification domain to human action recognition, constructing a new framework able to combine distinct features in a straightforward pipeline. *Second*, we propose simple yet efficient spatial and temporal local features from subgroups of joints, aggregated into global representations, allowing further learning to extract relevant information for classification. *Third*, we demonstrate the individual improvements of each step of the proposed framework by extensive experiments, showing that all parts are important to the final results. With these contributions we are able to provide state-of-the-art performance at very high speed on three well know datasets. In addition, the source code of this work is publicly available [1].

The rest of this chapter is organized as follows. In section 3.2, we present the proposed framework. The experimental evaluation of our method is presented in section 3.3 and our conclusions are presented in section 3.4.

3.2 Proposed Framework

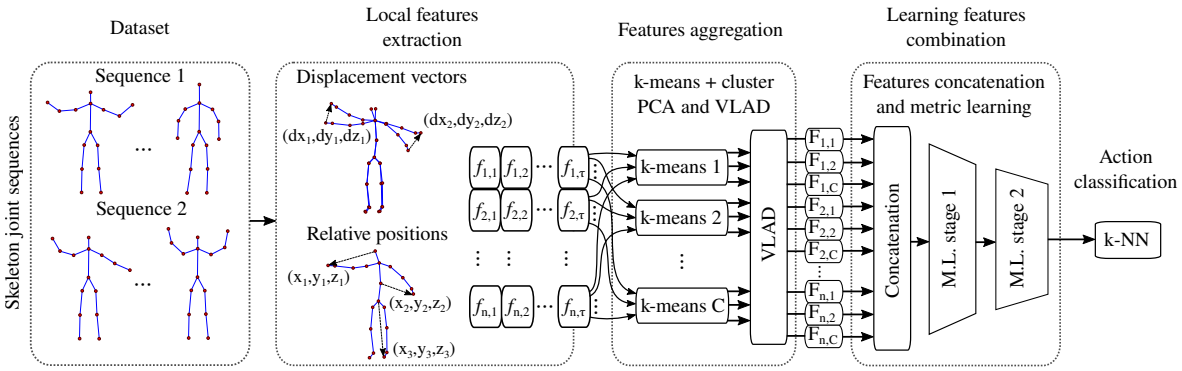


Figure 3.2 – Overview of the proposed framework for human action recognition from skeleton sequences.

The proposed approach for human action recognition rely on sequences of skeleton joints extracted from depth maps and can be divided into four stages, as outlined in Figure 3.2. In the first stage we extract local features, which are then aggregated into global representations in the second stage. In the third stage, all the resulting features are concatenated. Finally, the learning method extract the relevant information for the k-NN classification.

3.2.1 Local Features Extraction

The local features are extracted directly from sequences of skeleton joints and can be divided into two types, according to their physical interpretation. The first are the *displacement vectors* of joints, which represent the motion of specific body parts. Displacement vectors are 3D vectors taken from single joints with respect to the sequence of skeletons at time $t = \{1, 2, \ldots, \tau\}$, defined

[1] The Matlab® source code is publicly available at https://github.com/dluvizon/harskel

as follows:

$$\upsilon_i^t = \frac{p_i^{t+1} - p_i^{t-1}}{\Delta T} \mid 1 < t < \tau, \tag{3.1}$$

where p_i^t is the coordinate (x, y, z) of the ith joint in the sequence at time t, ΔT is the time interval between two skeletons at time $t+1$ and $t-1$, and τ is the maximum number of skeletons (frames) in a given sequence. The second type of local features are formed by *relative position* of joints, which is a relevant information that describes the body position and has been successfully used by other authors [148, 82]. The relative position between two joints in the skeleton sequence at time t is a 3D vector defined by the equation bellow:

$$\omega_{i,k}^t = p_i^t - p_k^t \mid i \neq k, \tag{3.2}$$

where p_i^t and p_k^t are the coordinates (x, y, z) of different joints (indexed by i and k) from the same skeleton.

The skeletal representation of human body is usually composed by a fixed number of joints. Two different layouts of human body representation are shown in Figure 3.3. The basic layout is composed by 15 joints: *right hand, r. elbow, r. shoulder, head, neck, left shoulder, l. elbow, l. hand, spine, r. hip, r. knee, r. foot, l. hip, l. knee*, and *l. foot*. Another representation commonly used in some datasets includes the *r. wrist, l. wrist, center hip, r. ankle*, and *l. ankle*, resulting in 20 joints. In order to build the proposed local features, we concatenate displacement vectors (from Equation 3.1) or relative positions (from Equation 3.2) taking subgroups of joints. Three features are composed by combination of displacement vectors and four features are composed by combination of relative positions, respectively detailed by Table 3.1 and Table 3.2. Specifically, the features f_1, f_2, and f_3 are composed by concatenation of displacement vectors of five joints. Similarly, the features f_4, f_5, f_6, and f_7 result from distinct relative positions concatenated together.

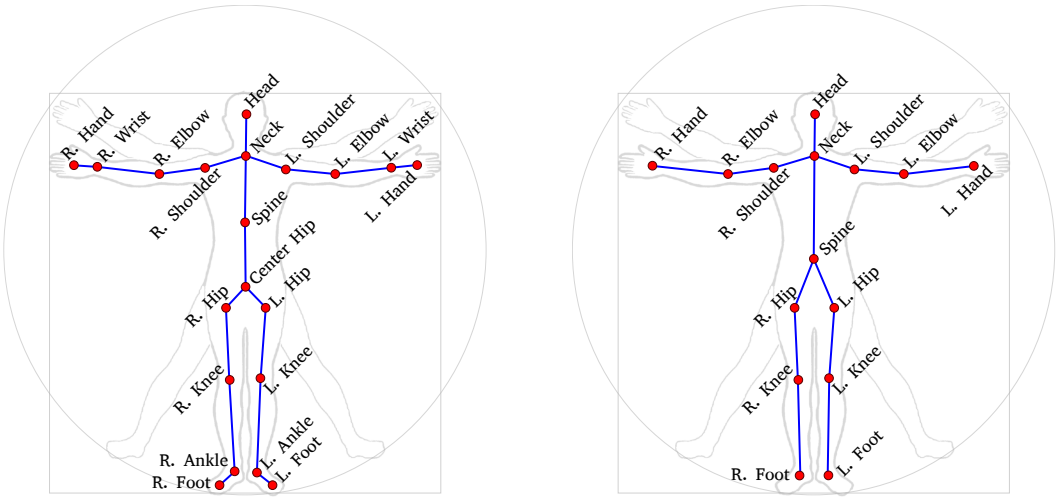


Figure 3.3 – Human body representation by 20 (a) and 15 (b) skeleton joints considering the datasets MSR-Action3D, UTKinect-Action3D, and Florence 3D Actions.

The major objective of dividing skeletons into subgroups of joints is to provide smaller features to the clustering stage. When compared to all joints composing one single feature, we expect two improvements. First, smaller features tend to be better clustered, and second, it is preferable to use smaller but complementary groups of joints than reducing the feature space (by PCA, for example). This assumption is verified by our experiments, as presented in section 3.3. Another important point is how the subgroups are chosen. In our method it is not practical to evaluate all

Table 3.1 – Features f_n composed by subgroups of displacement vectors.

Feature	Subgroup of joints (i)
f_1	*Head, r. hand, l. hand, r. foot,* and *l. foot*
f_2	*Neck, r. elbow, l. elbow, r. knee,* and *l. knee*
f_3	*Spine, r. shoulder, l. shoulder, r. hip,* and *l. hip*

Table 3.2 – Features f_n composed by subgroups of relative positions

Feature	Subgroup of joints (i)	Relative to (k)
f_4	*Head, l. hand,* and *r. hand*	*Spine*
f_5	*Head, l, hand,* and *l. foot*	*R. hip*
f_6	*Head, r. hand,* and *r. foot*	*L. hip*
f_7	*L. hand* and *r. hand*	*Head*

the possible combinations of different joints into smaller groups. Therefore, we intuitively divided the joints of displacement vectors from the center to the extremities of the human body. Similarly, the relative positions were selected to represent the position of hands, feet and head with respect to the rest of the body. This approach was empirically validated as a satisfactory solution by our experiments.

3.2.2 Features Aggregation

For each video frame, a set of local features are extracted by the local feature extraction method. Namely, the sequence of local features is represented by $f_{n,t}$ where $n = \{1,2,\ldots,7\}$ and $t = \{1,2,\ldots,\tau\}$. The objective of the aggregation stage is to build fixed-size features for each sequence of local features, as depicted in Figure 3.4.

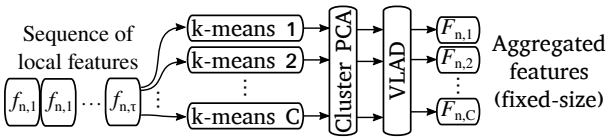


Figure 3.4 – Diagram of the feature aggregation stage.

The feature aggregation method can be divided into three steps. In the first step, for each subgroup n we compute C different k-means using different initializations. For each subgroup we then obtain C sets of K clusters represented by $\{\mu_{n,c,m}\}$, where $m = \{1,\ldots,K\}$ and $c = \{1,\ldots,C\}$. The next step consists in applying PCA to local features individually in each cluster. In this step, we apply PCA keeping all the components. Finally, in the third step, we use the vector of locally aggregated descriptors (VLAD), which is the non probabilistic version of the Fisher Vectors, proposed by [61]. Let

$$S_{n,c,m} = \left\{ f_{n,t} \,\middle|\, m = \arg\min_{p} \| f_{n,t} - \mu_{n,c,p} \| \right\} \tag{3.3}$$

be the set of local features of subgroup n from the initialization c in cluster m. Then the VLAD component m with respect to the initialization c is:

$$v_{n,c,m} = \sum_{f_{n,t} \in S_{n,c,m}} (f_{n,t} - \mu_{n,c,m}). \tag{3.4}$$

The VLAD representation of subgroup n and k-means c is simply the concatenation of all compo-

nents $v_{n,c,m}$, as follows:

$$\mathrm{F}_{n,c} = [v_{n,c,1}, \ldots, v_{n,c,\mathrm{K}}]. \tag{3.5}$$

As proposed by [62], we apply power law normalization keeping the sign of each component x of the VLAD representation by doing $x \leftarrow sign(x)\sqrt{|x|}$. The PCA inside clusters followed by the power law normalization can be seen as a kind of "whitening" [32]. As noted by [119], whitening vectors is equivalent to replacing the Euclidean distance by the Mahalanobis distance.

Each feature $\mathrm{F}_{n,c}$ is a vector which size depends only on the corresponding local feature size and the number of centers in the clustering algorithm. We do a flat concatenation of all features $\mathrm{F}_{n,c}$ into a final feature vector here represented by $\vec{x}$. The key factor in the aggregation method is that the fundamental structure of local features are preserved while using multiple clustering representations. That fact allows the next stage to learn the best combination of features and clustering representations.

3.2.3 Learning Features Combination

As a result from the previews two stages, we have a feature vector $\vec{x}$ formed by aggregated features that depends on the number of local feature subgroups (n) and the number C of unique k-means initializations. The goal of the feature combination stage is to extract discriminant information that improves the action recognition accuracy, considering the nonparametric k nearest neighbor (k-NN) classifier. In this regard, we employ two stages of metric learning, resulting in a reduced final feature vector, which is used for classification.

The metric learning approach is inspired by the large margin nearest neighbor (LMNN) algorithm proposed by Weinberger and Saul [151]. For simplicity, we define the squared distance (squared l2-norm) between two feature vectors, $\vec{x}_i$ and $\vec{x}_j$, in function of the linear transformation $\mathbf{L}$:

$$\mathscr{D}_{\mathbf{L}}(\vec{x}_i, \vec{x}_j) = \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 \tag{3.6}$$

3.2.3.1 Loss Function

Considering the k-NN classifier, the loss function is a measurement of violations made by *impostor* samples and distancing among *target* neighbors. Specifically, given a feature vector $\vec{x}_i$, the target neighbors, here represented by $\vec{x}_j$, are those that we want to be closest to $\vec{x}_i$. On the other hand, the *impostors*, represented by $\vec{x}_l$, are those that are closer to $\vec{x}_i$ without being targets. This concept was previously introduced by [151]. The function loss can be represented by two forces: the *pull* and *push* components, trying to respectively pull the targets while pushing the impostors, defined as follows:

$$\varepsilon_{pull}(\mathbf{L}) = \sum_{j \to i} \mathscr{D}_{\mathbf{L}}(\vec{x}_i, \vec{x}_j) \tag{3.7}$$

$$\varepsilon_{push}(\mathbf{L}) = \sum_{i,j \to i} \sum_{i,l \not\to i} [\xi + \mathscr{D}_{\mathbf{L}}(\vec{x}_i, \vec{x}_j) - \mathscr{D}_{\mathbf{L}}(\vec{x}_i, \vec{x}_l)] \tag{3.8}$$

where ξ is the desired separation margin between targets and impostors. The notations $j \to i$ and $l \not\to i$ mean that j is the index of targets of sample i while l is the index of impostors of sample i.

As many of the datasets for human action recognition have a relatively small number of samples from each action, learning algorithms can be very prone to *overfitting* on such data. To cope with this, we added a regularization in the linear transformation $\mathbf{L}$, as presented in the global loss

function:

$$\varepsilon(\mathbf{L}) = (1-\mu)\varepsilon_{pull}(\mathbf{L}) + \mu\varepsilon_{push}(\mathbf{L}) + \gamma\|\mathbf{L}^{\mathrm{T}}\mathbf{L} - \mathbf{I}\|^2 \tag{3.9}$$

where μ is the ratio between "push" and "pull" components, γ is the regularization coefficient, and $\mathbf{I}$ is the identity matrix. The regularization term enforces that the equivalent metric $\mathbf{L}^{\mathrm{T}}\mathbf{L}$ should remain close to the identity matrix.

3.2.3.2 Global Optimization

The optimal transformation $\mathbf{L}^*$ that minimizes Equation 3.9 can be found by solving the global optimization problem:

$$\mathbf{L}^* = \arg\min_{\mathbf{L}} \varepsilon(\mathbf{L}) \tag{3.10}$$

In order to solve Equation 3.10 using the gradient descent approach, we compute the derivative term of ε in $\mathbf{L}$, as follows:

$$\begin{aligned}\frac{1}{2}\frac{\partial\varepsilon}{\partial\mathbf{L}} = (1-\mu)\mathbf{L}\sum_{i,j\rightarrow i}(\vec{x}_i-\vec{x}_j)(\vec{x}_i-\vec{x}_j)^{\mathrm{T}} + \mu\mathbf{L}\sum_{i,j\rightarrow i}\sum_{i,l\not\rightarrow i}[(\vec{x}_i-\vec{x}_j)(\vec{x}_i-\vec{x}_j)^{\mathrm{T}} \\ -(\vec{x}_i-\vec{x}_l)(\vec{x}_i-\vec{x}_l)^{\mathrm{T}}] + 2\gamma\mathbf{L}(\mathbf{L}^{\mathrm{T}}\mathbf{L}-\mathbf{I})\end{aligned} \tag{3.11}$$

Since the number of operations required to solve Equation 3.11 can be significantly large even for small training datasets, we employ a minimization algorithm based on stochastic gradient descent (SGD) [12]. Let us define $\mathbb{D}$ as the training dataset. In the SGD optimization, for each iteration, we randomly select a small subset from $\mathbb{D}$ defined as $\mathbb{S}$. Iterating over the samples in $\mathbb{S}$, i.e., the index i is restricted to samples in $\mathbb{S}$, we solve Equation 3.11 taking targets and impostors from the whole dataset $\mathbb{D}$. A good initialization of $\mathbf{L}$ can be done by taking the eigenvectors of the covariance matrix of $\mathbb{D}$, which means to initialize $\mathbf{L}$ with the PCA on $\mathbb{D}$. This is a good initialization since we reduce the feature size in the metric learning stages and PCA is known to be a good dimension reduction technique. The optimization is performed until the maximum number of epochs (*MaxEpoch*) is reached or the gradient vanishes according to the threshold ϑ. The global SGD optimization is presented in 1.

Algorithm 1 Global SGD optimization.

Require: Training dataset $\mathbb{D}$, $MaxEpoch$, vanishing value ϑ
1: Do PCA on $\mathbb{D}$ to initialize $\mathbf{L}$
2: $Epoch \leftarrow 0$
3: **repeat**
4: $\quad\mathbb{S} \leftarrow$ Randomly select samples from $\mathbb{D}$
5: $\quad\mathbf{G} \leftarrow$ Solve Equation 3.11 for the subset $\mathbb{S}$
6: $\quad\mathbf{L} \leftarrow \mathbf{L} - \eta\mathbf{G}$
7: $\quad Epoch \leftarrow Epoch + sizeof(\mathbb{S})/sizeof(\mathbb{D})$
8: **until** $(\|\mathbf{G}\| \geq \vartheta)$ **and** $(Epoch < MaxEpoch)$
9: **return** $\mathbf{L}$

As shown in Figure 3.2, two stages of metric learning are used in our method. The first one aim to reduce the feature size while performing a first separation between targets and impostors by learning the transformation $\mathbf{L_1}$. In this regard, we set $\mu = 0.9$ and $MaxEpoch = 2$. The second stage works on smaller features and learns the transformation $\mathbf{L_2}$ with $\mu = 0.5$ and $MaxEpoch = 50$. Each metric learning stage is individually optimized following the Algorithm 1, replacing $\mathbf{L}$ by $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively at each etage. Namely, we first learn $\mathbf{L_1}$, which takes $\vec{x}$ as input, then in

the second stage we learn $\mathbf{L_2}$ which input is $\mathbf{L_1}\vec{x}$. This approach allows a fast learning process in addition to avoiding overfitting, since the feature size is reduced in a few iterations and the more intensive learning stage is performed over fewer parameters. Finally, since all transformations are linear, in the testing evaluation we can use $\mathbf{L} = \mathbf{L_2}\mathbf{L_1}$ to represent the full learned transformation.

3.3 Experiments

We evaluated the accuracy of our method on three publicly available datasets. The MSR-Action3D [75] is the most common dataset for 3D human action recognition according to [167] and is composed by 10 subjects performing 20 actions chosen in the context of gaming, which include: *high wave, horizontal wave, hammer, hand catch, high throw, draw X, draw tick, draw circle, hand clamp, two hand clamp, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing*, and *pick-up throw*. This dataset is challenging due to some very similar pairs of actions, for example: *hand catch* and *draw tick*, or *pick-up throw* and *bend*. The UTKinect-Action3D dataset [153] is composed by 10 subjects, of which nine are males and one is female including one left-handed, performing 10 actions: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave*, and *clap hands*. Each subject perform actions in various views and the length of videos vary from 5 to 120 frames, resulting in significant variation among the recordings. The Florence 3D Actions dataset [122] is composed by 10 subjects performing 9 actions recorded in distinct environment conditions, which include: *wave, drink from a bottle, answer phone, clap, tight lace, sit down, stand up, read watch*, and *bow*. Since our feature extraction method requires only 15 skeleton joints, we averaged the joints from hands and wrist into a single joint *hand* for the datasets MSR-Action3D and UTKinect-Action3D, in which skeletons are composed by 20 joints. We apply the same process to foot and ankle, and to spine and center hip.

For all datasets, we use the already computed skeleton joints data and the same parameters in all the performed tests. We optimized the hyperparameters of our method using only the MSR-Action3D dataset split as proposed by [148]. Seven local feature subgroups were extracted as described in section 3.2.1. In the feature aggregation stage, we use a pool of five unique k-means (C = 5), each one computing K = 23 clusters. After the feature concatenation stage, the resulting feature vectors are of size 8970. In both metric learning stages, we set $\gamma = 0.1$ and $\xi = 0.1$. In the SGD optimization, we solve the Equation 3.11 by taking batches of 32 training samples. In the first metric learning stage we set the output dimension to 512, followed by the second stage with output dimension equals to 256, which is the final feature size. The final classification is a seven nearest neighbors voting.

3.3.1 Comparison with the State of the Art

We compared our results with several methods in the state of the art on three distinct datasets, as presented in Table 3.3.

3.3.1.1 MSR-Action3D Dataset

The MSR-Action3D dataset has been used by several works in many disparate ways. In our tests, we selected the two most relevant evaluation approaches on this dataset. The first approach we used is the cross-subject splitting proposed by [148], where subjects 1,3,5,7,9 are used for training and subjects 2,4,6,8,10 are used for testing. In that case, the accuracy of our method is 97.1%, which is the best result on this data as far as we know. Comparable results are shown in Table 3.3.

Table 3.3 – Accuracy evaluation of our method compared to the state-of-the-art methods on three public datasets. Columns two and three present results on the MSR-Action3D dataset using the protocols proposed by [148] and [96], respectively. Columns four and five respectively show results on UTKinect-Action3D and Florence 3D Actions datasets.

Dataset / Method	MSR-Action3D protocol of [148]	MSR-Action3D protocol of [96]	UTKinect-Action3D	Florence 3D Actions
Wang et al. [148]	88.2%	—	—	—
Xia et al. [153]	—	—	90.92% ± 1.74%	—
Luo et al. [82]	96.7%	—	—	—
Oreifej and Liu [96]	88.36%	82.15% ± 4.18%	—	—
Seidenari et al. [122]	—	—	—	82.0%
Tran and Ly [140]	88.89%	—	—	—
Devanne et al. [33]	92.1%	87.28% ± 2.41%	91.5%	87.04%
Lu et al. [81]	95.62%	—	—	—
Vemulapalli et al. [144]	89.48%	—	97.08%	90.88%
Yand and Tian [159]	93.09%	—	—	—
Slama et al. [128]	91.21%	—	88.5%	—
Veeriah et al. [143]	92.03%	—	—	—
Li and Leung [72]	92.2%	—	—	—
Rahmani et al. [112]	—	86.5%	—	—
Our method	**97.1%**	**90.36% ± 2.45%**	**98.00% ± 3.49%**	**94.39%**

As can be seen in the confusion matrix (see Figure 3.5) resulting from our method, several actions were classified without any mistake and only two actions presented classification accuracy lower than 93%. The second approach for evaluation we used was proposed by [96], where we report the average result among all possible 5-5 subject splits. We consider this approach the most relevant, since it reduces the possibility of effects from particular combinations. By this approach, our method achieved an average accuracy of 90.36% and a standard deviation of 2.45%, which is an improvement of 3.08% over the best method so far [33]. We reinforce that in both approaches results are reported in the cross-subject scenario. The results from other methods using the same assessment are reported in the third column of Table 3.3.

3.3.1.2 UTKinect-Action3D Dataset

On this dataset, the authors [153] proposed to use the leave one actor out cross validation (LOOCV) scheme. Specifically, one actor is removed from training and used as testing. This process is repeated for all actors and the final result is the average accuracy of all runs. On our tests, we followed the same procedure and our method achieved on average 98.00% of accuracy with a standard deviation of 3.49%, as reported in the fourth column of Table 3.3. We consider the LOOCV scheme statistically more stable than the single cross-validation assessment employed by [175] and [72]. Therefore, our results are not comparable on this dataset.

3.3.1.3 Florence 3D Actions Dataset

Similarly to the previews experiment, we evaluate the performance of our method on the Florence 3D Actins dataset using the LOOCV approach, as suggested by the authors [122]. On average, our method classified 94.39% of actions correctly. Our method exceeded the state-of-the-art approaches by a significant margin, as presented in the fifth column of Table 3.3.

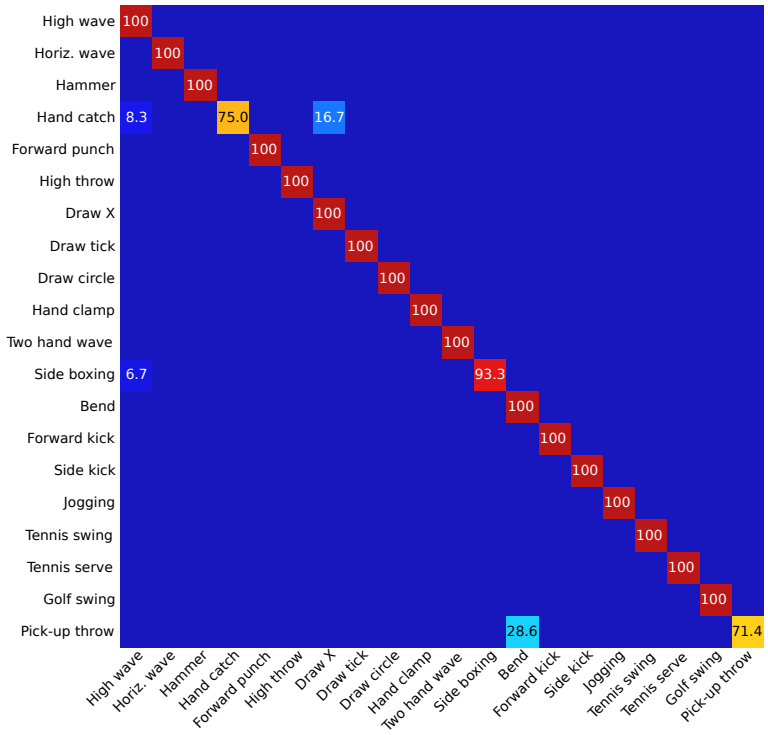


Figure 3.5 – Confusion matrix for action classification on the MSR-Action3D dataset resulted from the proposed method.

3.3.2 Contribution of each Method's Stage

In this section, we discuss the influence of each part of our method, as tested on the MSR-Action3D dataset.

- First, using all joints together instead of our proposed subgroups leads to a performance decrease of 5.5%.
- If only displacement vectors or only relative positions are used, the classification accuracy drops by 4.1% and 17.2%, respectively.
- In the feature aggregation stage, if the PCA or the power law normalization is turned off, the performance decreases by 4.4% and 4.8%, respectively.
- Similarly, aggregating features with a single clustering initialization, i.e., setting C = 1, drops the performance by 2.5%.
- Replacing the proposed two stages of metric learning by features reduction with PCA, it means using PCA to reduce the feature size from 8970 to 512 and then using a single metric learning stage, the best performance decreases by 0.8%.
- Removing the regularization coefficient from Equation 3.9 reduces the best performance by 0.4% and led to faster overfitting.
- Finally, replacing the k-NN classifier by SVM or neural network (MLP) drops the performance by 1.5% in the best case (see Table 3.4).

The conception of the proposed framework was reasoned that each part is optimally designed regarding the next stage in the pipeline. For instance, the local features extraction provides small

features that can be clustered well, while avoiding early combination of distinct information. In the clustering stage, we use multiple initializations to increase the chances to have a better representation, which can be learned in the next stage. Similarly, the metric learning algorithm (LMNN) is optimal to increase the nearest neighborhood (k-NN) classifier accuracy.

The multiple clustering initialization is an important step in the feature aggregation method and goes beyond the improvement on classification accuracy of the proposed framework. As shown in Figure 3.6, the probability of reaching better accuracy drastically increases after metric learning when using C = 5. This effect can also be observed by the standard deviation decreasing from 1.34% to 0.71%, respectively using C = 1 and C = 5. This fact is expected, since the metric learning can extract complementary information from different clustering representations. Additionally, the metric learning stage can be seen as the point of convergence where all the particular improvements are intensified, resulting in a final improvement of 12% as shown in Figure 3.7, while drastically reducing the feature size.

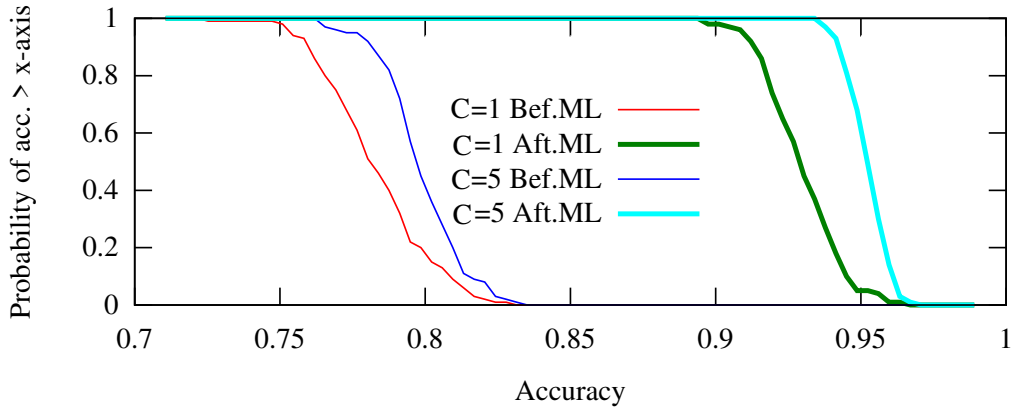


Figure 3.6 – Probability of accuracy according to 100 random evaluations using C = 1 and C = 5 in the feature aggregation stage, before (Bef.ML) and after (Aft.ML) metric learning.

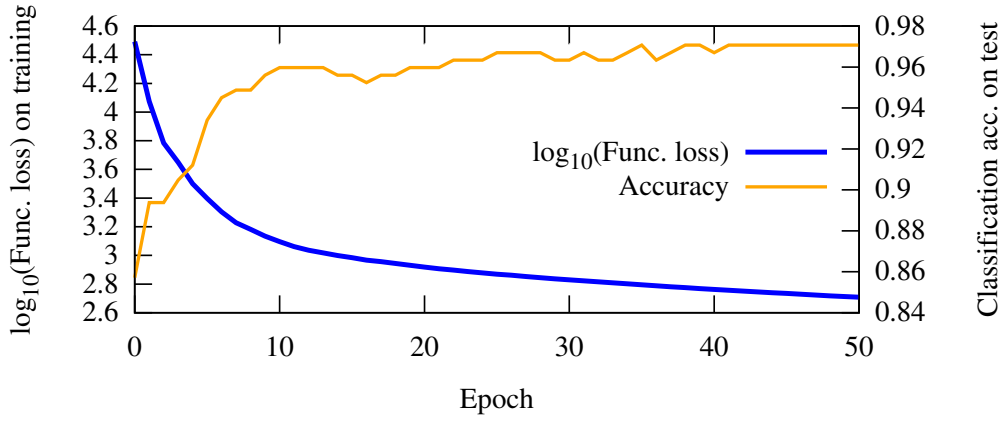


Figure 3.7 – The learning curves: evolution of the function loss on training and the classification accuracy on testing samples.

We evaluate the influence of the last stages by replacing the k-NN classifier by two well known classifiers, before and after the metric learning (LMNN) stage. First, we compared with a standard SVM [18] using the sigmoid kernel setting the parameters *gamma* and *C* respectively to 1 and 10. Second, we compared with a neural network (MLP) with two fully-connected layers, the first using *ReLu* activation and the second using *softmax* for classification. Table 3.4 shows that the k-NN

classifier on learned features by the metric learning stage gives the best result, even though both SVM and neural network present results comparable to the state of the art, which demonstrates the robustness of the proposed features representation. These are expected results, since the objective function of the metric learning stage was specially designed to increase the k-NN classification accuracy.

Table 3.4 – Classification accuracy of the proposed metric learning and classifier stages compared to SVM and neural network approaches. We evaluate the classifiers taking as input the aggregated features ($\vec{x}$) and the learned features after LMNN.

Classifier	Aggregated features	Learned features (LMNN)
SVM	93.4%	95.6%
Neural net.	93.8%	95.6%
k-NN	83.2%	**97.1%**

3.3.3 Computation Time

The average testing runtime of the proposed method is presented in Table 3.5, which is faster than the computation time reported by [128]. The testing sequences were processed in a laptop machine with Intel® Core™ i7-4710MQ processor, after training. One of the reasons which led to low computing time for action recognition is that the most complex part of our method is the metric learning feature combination. Once the training stage is finished, recognizing new sequences is a fast and straightforward process.

Table 3.5 – Average testing runtime of the proposed method on three datasets. The given computation time in milliseconds refers to one testing sequence.

Dataset / Stage	MSR-Action3D	UTKinect Action3D	Florence 3D Actions
Local features extraction (ms)	2.34	2.00	1.52
Features aggregation (ms)	4.59	4.92	4.14
Features combination (ms)	0.14	0.18	0.15
Classification k-NN (ms)	1.38	0.97	1.06
Average testing time (ms)	**8.45**	**8.07**	**6.87**

3.4 Conclusion

In this chapter, we presented a new framework for human action recognition using only skeleton joints extracted from depth maps. We proposed extracting sets of spatial and temporal local features from subgroups of joints. Local features are aggregated into several feature vectors by a robust method using the VLAD algorithm and a pool of clusters, providing a good representation for long and short actions. All the feature vectors are then efficiently combined by a metric

learning method inspired by the LMNN algorithm, which is used to extract the most discriminant information from features with the objective to improve the accuracy of the k-NN classifier.

Extensive experiments with the proposed framework show that all the proposed steps contribute significantly to improve classification accuracy. We conclude that spatial and temporal information, as well as the multiple clustering representations, could be efficiently combined by the metric learning approach, resulting in a significant increase of performance. Moreover, the proposed method relies on a few external parameters and our experiments show that the method generalizes well, since its performance overcame all the results in the state of the art on three important datasets, using the same parameters in all evaluations.

Chapter 4

Human Pose Estimation from RGB Images

Contents

Prologue

Related Articles:

- D. C. Luvizon, H. Tabia, D. Picard. **Human Pose Regression by Combining Indirect Part Detection and Contextual Information**. *CoRR, abs/1710.0232*, **pre-print**, 2017.
- D. C. Luvizon, D. Picard, H. Tabia. **2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning**. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5137-5146, 2018.
- D. C. Luvizon, H. Tabia, D. Picard. **SSP-Net: Scalable Sequential Pyramid Networks for Real-Time 3D Human Pose Regression**. **pre-print**, 2018.

Context:

In chapter 3, we presented a new framework for action recognition from skeleton sequences. A common way to predict human poses or skeletons is by analysing depth images captured by commodity depth sensors, such as the Microsoft's Kinect. However, such devices are limited to controlled environment, have a low range of operation, and drastically suffer from occlusions, resulting in very noisy skeleton predictions when exposed to non-optimal conditions. These limitations associated to the recent breakthrough of deep learning is a strong motivation to study human pose estimation from RGB images, since nowadays such images are widely available, including in unconstrained environments, on which recent deep learning algorithms have demonstrated satisfactory results even in challenging scenarios.

In this chapter, we present a study about the problem of human pose estimation from still RGB images, which is divided in three parts, detailed as follows.

In the *first* part, we present a regression approach, trainable from end-to-end, for 2D keypoints regression based on the *soft-argmax* function. In the proposed regression method, feature maps are directly converted to joint coordinates, resulting in a fully differentiable framework. Our method is able to learn heat map representations indirectly, without additional steps of artificial ground truth generation. Consequently, contextual information can be included to the pose predictions in a seamless way. We also demonstrate the generalization if this method to 3D predictions by learning volumetric heat map representations. Our method is evaluated on two very challenging datasets, the Leeds Sports Poses (LSP) and the MPII Human Pose datasets, reaching the best performance among all the existing regression methods and comparable results to the state-of-the-art detection based approaches.

In the *second* part, we present a new highly scalable network architecture for real-time 3D human pose regression from RGB images. This new architecture, called Scalable Sequential Pyramid Networks (SSP-Net), predicts human poses directly in 3D coordinates and is trained with dense supervision at multiple scales. The SSP-Net is capable of producing its best predictions at 120 fps, or acceptable predictions at more than 200 fps, while requiring a single training procedure. We also propose a new 3D pose regression attention mechanism based on regressed 2D depth maps, departing from the expensive volumetric heat map representations. The proposed regression approach is invariant to the size of feature maps, allowing our method to perform multi-resolution intermediate supervisions and reaching results comparable to the state-of-the-art with very low resolution feature maps. We demonstrate the accuracy and the effectiveness of our method by providing extensive experiments on two of the most important publicly available datasets for 3D pose estimation (Human3.6M and MPI-INF-3DHP). Additionally, we provide relevant insights about

our decisions on the network architecture and show its flexibility to meet the best precision-speed compromise.

Finally, in the *third* part, we extend our regression method to absolute 3D human pose estimation in real world coordinates. Our approach predicts several 3D pose proposals and estimates the absolute depth of each root joint. We also propose new skeleton template composed of 34 body joints, which integrates several recent human pose datasets with no ambiguity for specific joint positions. Our method consistently improves the state of the art on well known 3D pose benchmarks, reducing prediction error by more than 25% in some cases. For the first time, we report results on 3D pose estimation on absolute world coordinates, showing comparable results to root joint centered metrics.

4.1 Introduction

Predicting human poses from still RGB images is a hard task since the human body is strongly articulated, some parts may not be visible due to occlusions or low image quality. Furthermore, the 3D world information is partially missing due to the image plane projection, and the visual appearance of body parts can change significantly from one person to another. Meanwhile, both 2D and 3D pose estimation problems have been intensely studied in the last years, mostly because of their promising applications, such as 3D scene understanding, sports performance analysis, style transfer, 3D model fitting, human behavior analysis, human action recognition, among others.

The human pose estimation problem can be cast into three different categories:

- *Pose estimation in the image plane*, or simply *2D pose estimation*, where 2D keypoints are represented in the format (x, y) and correspond to image pixel coordinates.
- *Relative 3D pose estimation*, or simply *3D pose estimation*, where each body joint is represented by its 3D coordinates (x, y, z) in millimeters relative to the root joint, which is usually taken as the *hip joint*. In this category, the coordinate system is centered in the target person.
- *Absolute 3D pose estimation*, on which predictions are also in millimeters, but differently from the case of relative prediction, the coordinate system is fixed in the world, which makes the problem much more challenging.

Classical methods for 2D pose estimation use keypoint detectors to extract local information, which are combined to build pictorial structures [38]. To handle difficult cases of occlusion or partial visualization, contextual information is usually needed to provide visual cues that can be extracted from a broad region around the part location [37] or by interaction among detected parts [160]. In general, 2D human pose estimation can be seen from two different perspectives, namely as a correlated part detection problem or as a regression problem. Detection based approaches commonly try to detect keypoints individually, which are aggregated in post-processing stages to form one pose prediction. In contrast, methods based on regression use a function to map directly input images to body joint coordinates.

In the last few years, pose estimation have gained special attention with the breakthrough of deep Convolutional Neural Networks (CNN) [139] alongside consistent computational power increase. This can be seen as the shift from classical approaches [104, 68] to deep architectures [90, 19]. In many recent works from different domains, CNN based methods have overcome classical approaches by a large margin [47, 127]. A key benefit from CNN is that the full pipeline is differentiable, allowing end-to-end learning. However, recent detection based methods for both 2D and 3D pose estimation [156, 22, 99] are based on heat maps prediction, which are then converted to coordinates by applying the maximum a posteriori (MAP) estimation, usually called *argmax*. Since the argmax is a non-differentiable operation, such methods are not trainable from end-to-end. This technique is used to cast pose estimation as a classification problem. Additionally to the non-differentiability, the precision of predicted body joints is proportional to that of the heat map resolution, which leads such approaches to high memory consumption and high computational requirements, specially in the case of 3D predictions.

An alternative to detection based approaches is to perform coordinates regression directly from images [139]. However, due to the high variance in both images and in the high articulated body skeleton, simple regression approaches such as fully connected layers are usually suboptimized, generally resulting in lower precision if compared to detection approaches. A solution

to this problem is to replace the non differentiable argmax by the expectancy of normalized heat maps, which in the literature is called *soft-argmax* [84] or *integral regression* [132]. This approach has also another advantage compared to predicting heat maps, which is the easy combination of 3D and 2D annotated data just by not propagating the error on z in the last case. As demonstrated in our experiments (see section 4.4), this has been proved a very efficient data augmentation technique, improving precision on 3D predictions by a significant margin.

Despite the recent progress on 3D human pose estimation, current methods still depend on the expensive 3D heat maps [132]. Moreover, the popular hourglass networks [93], very common on pose estimation methods, are not designed to consider predictions at multiple scales, mainly because detection based approaches require high resolution heat maps. We tackle these limitations by proposing a new 3D regression approach based on 2D depth map prediction and a new multiscale network architecture, as detailed in section 4.5.

Furthermore, predicting the absolute 3D body joints in world coordinates from monocular RGB images is still an open problem. Precise 3D human pose estimation in world coordinates is still dependent on expensive Motion Capture (MoCap) systems, which require complex calibration procedures making it prohibitive in several uncontrolled environments. State-of-the-art methods for relative 3D pose estimation are incapable of predicting the body joints position in real world coordinates, because they are designed for handling the much easier root joint centered coordinate system. However, the absolute position can be very useful in many applications, where disambiguating the relative depth of multiple people is essential. Additionally, predicting pose coordinates in the world reference also provides an advantage on multi-view scenarios, since predictions from different views could be easily combined.

Inspired by the exposed limitations of current methods for human pose estimation, we present the main contributions of this chapter as follows.

- First, we present a new human pose regression approach from still images based on the soft-argmax function, resulting in a method trainable from end-to-end which does not require artificial heat maps generation and can be trained with an insightful regression loss function by directly linking the error distance between predicted and ground truth joint positions.
- Second, we propose to learn depth map representations for 3D pose regression, departing from the required and expensive 3D heat maps and achieving state-of-the-art results on 3D human pose estimation while reducing computations.
- Third, we propose the Scalable Sequential Pyramid Networks, which is a new, fast, and efficient CNN architecture, producing high precise 3D human pose predictions at more than 100 FPS, while being easily scalable to perform faster predictions up to 300 FPS with a single training procedure.
- Fourth, we propose a method for 3D human pose estimation on absolute world coordinates and we are the first to report results on absolute error comparable to root joint centered errors, when considering multi-view scenario.
- Fifth, thanks to a new skeleton layout and to the ability of our method to combine multiple-camera predictions, we are able to improve 3D human pose estimation by more than 25% in a very challenging dataset.

The remaining of this chapter is divided as follows. In section 4.2, we present the proposed keypoints regression approach based on the soft-argmax operation. In section 4.3, we present a

study on 2D pose estimation. The more complex case of 3D relative pose prediction is detailed in section 4.4. In section 4.5, we proposed an improved 3D pose regression approach based on learned depth maps and a scalable network architecture for more precise and efficient estimations. The challenging problem of absolute pose prediction is addressed in section 4.6. Finally, in section 4.7 we present our conclusions for this chapter.

4.2 Differentiable Keypoints Regression

As presented in section 4.1, traditional regression based methods use fully connected layers on feature maps to learn the regression mapping to pose coordinates. However, this approach usually gives sub-optimal solutions. While state-of-the-art methods on 2D pose estimation are overwhelmingly based on part detection, approaches based on regression have the advantage of providing directly pose prediction as joint coordinates without additional steps or post-processing. In order to provide an alternative to detection based methods, we propose an efficient and fully differentiable way to convert heat maps to (x, y) coordinates, which is called *soft-argmax*. The idea was previously used for features extraction [163], but, to the best of our knowledge, we are the first to propose soft-argmax for human pose estimation, as detailed in our previous work [84].

Given an input signal, the main idea is to consider that the argument of the maximum 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. As detailed in section 4.2.1, the normalized exponential function (softmax) is used, since it alleviates the undesirable influences of values bellow 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 of the normalized signal, as detailed in section 4.2.2.

4.2.1 Spatial Softmax

Let us redefine the softmax operation on a single heat map $\mathbf{h} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ as:

$$\Phi(\mathbf{h}) = \frac{e^{\mathbf{h}}}{\sum_{l=1}^{\mathrm{H}}\sum_{c=1}^{\mathrm{W}} e^{\mathbf{h}_{l,c}}}, \tag{4.1}$$

where $\mathbf{h}_{l,c}$ is the value of $\mathbf{h}$ at location (l, c) and $\mathrm{H}\times\mathrm{W}$ is the heat map size. Contrarily to the more common cross-channel softmax, we use here a spatial softmax to ensure that each heat map $\mathbf{h}$ is normalized (positive and unitary sum), additionally working as a non-maximum suppression. The normalized heat map is called joint *probability map* and is defined by:

$$\mathbf{h}' = \Phi(\mathbf{h}) \tag{4.2}$$

4.2.2 Soft-argmax for 2D Regression

Considering the probability map $\mathbf{h}'$ as input, the soft-argmax for 2D regression is defined as follows:

$$\Psi_d(\mathbf{h}') = \sum_{i=1}^{\mathrm{H}}\sum_{j=1}^{\mathrm{W}} \mathbf{W}_{d,i,j}\mathbf{h}'_{i,j}, \tag{4.3}$$

where d is a dimension component x or y, and $\mathbf{W}$ is a $2 \times \mathrm{H} \times \mathrm{W}$ weight matrix for both components (x, y). The matrix $\mathbf{W}$ can be expressed by its components $\mathbf{W}_x$ and $\mathbf{W}_y$, which are 2D discrete normalized ramps, defined as follows:

$$\mathbf{W}_{x,i,j} = \frac{2j-1}{2\mathrm{W}}, \mathbf{W}_{y,i,j} = \frac{2i-1}{2\mathrm{H}}. \tag{4.4}$$

Finally, given a heat map $\mathbf{h}$, the regressed position in 2D coordinates of the predicted joint is given by:

$$\hat{\mathbf{p}}_{2d} = \left(\Psi_x(\mathbf{h}), \Psi_y(\mathbf{h})\right)^{\mathrm{T}}. \tag{4.5}$$

The soft-argmax operation can be seen as the 2D expectation of the normalized heat map, which is a good approximation of the argmax function, considering that the exponential normalization results in a pointy distribution.

In order to integrate the soft-argmax into a deep neural network layer, we need its derivative with respect to $\mathbf{h}$:

$$\frac{\partial \Psi_d(\mathbf{h})}{\partial \mathbf{h}_{i,j}} = \mathbf{W}_{d,i,j}\Phi(\mathbf{h})_{i,j}(1-\Phi(\mathbf{h})_{i,j}) - \sum_{l=1}^{\mathrm{H}}\sum_{c=1}^{\mathrm{W}} \mathbf{W}_{d,l,c}\Phi(\mathbf{h})_{i,j}\Phi(\mathbf{h})_{l,c}\Big|_{l\neq i; c\neq j} \tag{4.6}$$

The soft-argmax function can thus be integrated with a trainable framework by using back propagation and the chain rule on Equation 4.6. Moreover, similarly to what happens on softmax, the gradient is exponentially increasing for higher values, resulting in very discriminative response at the joint position.

An intuitive graphical explanation of the soft-argmax is shown in Figure 4.1. Unlike traditional argmax, soft-argmax provides sub-pixel accuracy, allowing good precision even with very low resolution. Our approach allows learning very discriminative heat maps directly from the (x, y) joint coordinates without explicitly computing artificial ground truth. In our experiments, we show samples of heat maps indirectly learned by our method.

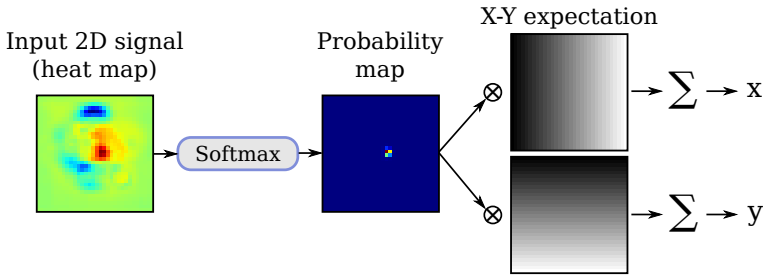


Figure 4.1 – Graphical representation of the soft-argmax operation for 2D input signals (heat maps). The outputs are the coordinates x and y that approximates the maximum in the input signal.

4.2.3 Confidence Score

As a complement to joint locations, we can compute the estimated joint probability $\hat{\mathbf{b}}_n$ and the joint confidence scores $\hat{\mathbf{c}}_n$ for the n^{th} body joint. The first corresponds to the probability of the joint being visible (or present, even if occluded) in the image and is defined by:

$$\hat{\mathbf{b}} = \frac{1}{1+e^{-h_{max}}}, \tag{4.7}$$

where h_{max} is the maximum value in the heat map $\mathbf{h}$. In other words, the joint probability $\hat{\mathbf{b}}$ is the sigmoid activation on the maximum response from the heat map. Since a heat map can have multiple "peaks", resulting in high joint probability score but wrong pose prediction, we define a second score to represent the confidence that the regressed position corresponds to the region with the maximum response in the heat map, as defined by:

$$\hat{\mathbf{c}} = max\left(\sum_{l=i}^{i+1}\sum_{c=j}^{j+1}\mathbf{h}'_{l,c}\right)\Bigg|_{i=\{1,\ldots,\mathrm{H}-1\};j=\{1,\ldots,\mathrm{W}-1\}}, \tag{4.8}$$

Specifically, given a joint probability map, any window with 2×2 pixels is enough to regress a coordinate value with sub-pixel accuracy in a smaller squared region defined by the centers of the 2×2 pixels, as depicted in Figure 4.2. Therefore, we apply a summation with a 2×2 sliding window on the probability map with stride 1, and take the maximum response as the confidence score. If the probability map is very pointy, the score is close to 1. On the other hand, if the probability map is smooth or has more than one region with high response, the confidence score drops.

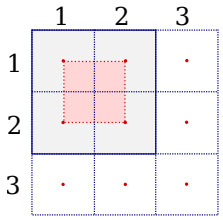


Figure 4.2 – Estimation of joint confidence scores. The blue squares represent the pixels in the joint probability map with its center marked as a red dot. The red square is the region on which a coordinate can be regressed, considering responses only on the 2×2 window from pixels (1, 1) to (2, 2).

Despite giving an additional information, the joint probability and confidence scores do not depend on additional parameters and are computationally negligible, compared to the cost of the convolutional layers. Additionally, by supervising these values, we can enforce the network to learn pointy responses for available body parts, which works as a constraint to the indirectly learned heat maps.

4.3 2D Human Pose Regression from RGB Images

In this section we present a 2D human pose estimation method from RGB images based on the regression approach introduced in section 4.2. Our implementation of the proposed method using the open source Keras library [26] is publicly available.[1]

The proposed approach is an end-to-end trainable network which takes as input RGB images and outputs two vectors: the probability $\hat{\mathbf{b}}_n$ of joint n being in the image and the regressed joint coordinates $\hat{\mathbf{p}}_n = (x_n, y_n)$, where $n = \{1, 2, \ldots, \mathrm{N_J}\}$ is the index of each joint and $\mathrm{N_J}$ is the number of joints in the human body layout. Thanks to the our regression approach based on the soft-argmax, we can learn two types of heat maps. The first, which we call *part-based* heat map, is specialized to respond specifically to each body joint. The second type is called *contextual* feature map and is considered in the final prediction based on its probability score, which means that if a specific contextual feature map does not respond to a given image, its influence in the final prediction will be attenuated.

[1] The Python source is publicly available for research purposes at https://github.com/dluvizon/pose-regression.

In what follows, we first present the global architecture of our method, and then detail its most important parts.

4.3.1 Network Architecture

An overview of the proposed method is presented in Figure 4.3. Our approach is based on a convolutional neural network essentially composed of three parts: the entry flow (stem), block-A and block-B. The role of the stem is to provide basic features extraction, while block-A provides refined features and block-B provides body-part and contextual activation maps. One sequence of block-A and block-B is used to build one *prediction block*, which output is used as intermediate supervision during training. The full network is composed by the stem and a sequence of K prediction blocks. The final prediction is the output of the K^{th} prediction block. To predict the pose at each prediction block, we aggregate the 2D coordinates generated by applying soft-argmax to the part-based and contextual maps that are output by block-B. Similarly to recent approaches [93, 29], we produce one estimation on each prediction block. This prediction is used as intermediate supervision, providing better accuracy and more stability to the learning process. As a convention, we use the generic term "heat map" to refer both to part-based and contextual feature maps, since these feature maps converge to heat maps like representations.

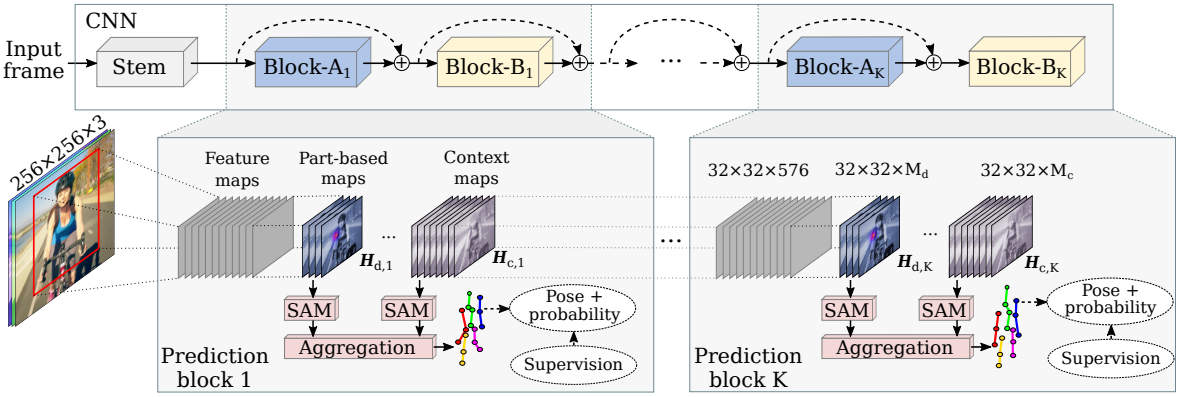


Figure 4.3 – Overview of the proposed method for 2D human pose regression. SAM: soft-argmax.

The proposed CNN model is partially based on Inception-v4 [133] and inspired by the Stacked Hourglass [93] networks. We also get some inspiration from the "extreme" inception (Xception) [25] networks, which relies on the premise that convolutions can be separated into spatial convolutions (individual for each channel) followed by a 1×1 convolution for cross-channel projection, resulting in a significant reduction on the number of parameters and on computations. This idea is called *depthwise separable convolution* (SepConv) and an optimized implementation is available on TensorFlow.

The "Stem" network is based on Inception-v4's stem followed by a SepConv layer in parallel with a shortcut layer, as presented in Figure 4.4a. The architecture of block-A is similar to an hourglass block, as proposed in [93], except that we replaced the residual blocks by a residual depthwise separable convolution (Res-SepConv), as depicted in Figure 4.4b, and reduced the number of internal scales from five to three, using feature maps from 32×32 to 8×8 instead of 64×64 to 4×4. The architectural details about these blocks can be consulted in Appendix A.

The architecture of block-B and the regression stage is shown in Figure 4.5. At each prediction stage, block-B is used to transform input feature maps into M_d *part-based detection maps* ($\mathbf{H}_d$) and M_c *context maps* ($\mathbf{H}_c$), resulting in $M = M_d + M_c$ heat maps. For the problem of human pose

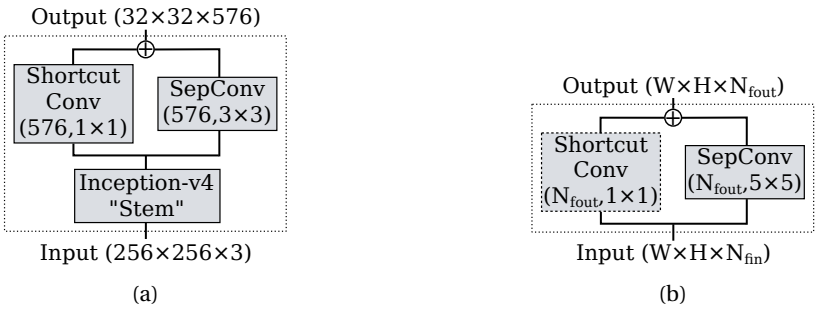


Figure 4.4 – In the proposed network architecture, the Stem (a) is based on Inception-v4's stem [133] followed by a separable convolution in parallel to a shortcut connection. In (b), we present the residual separable convolution (Res-SepConv), used to replace the residual block in the Stacked Hourglass [93] model. If N_{fin} is equal to N_{fout}, the shortcut convolution is replaced by the identity mapping.

estimation, M_d corresponds to the number of joints N_J, and $M_c = N_c N_J$, where N_c is the number of context maps per joint. The produced heat maps are projected back to the feature space and reintroduced to the network flow by a 1×1 convolution. Similar techniques have been used by many previous works [13, 93, 29], resulting in significant gain of accuracy. From the generated heat maps, our method computes the predicted joint locations and joint probability scores in the regression block, which has no trainable parameters.

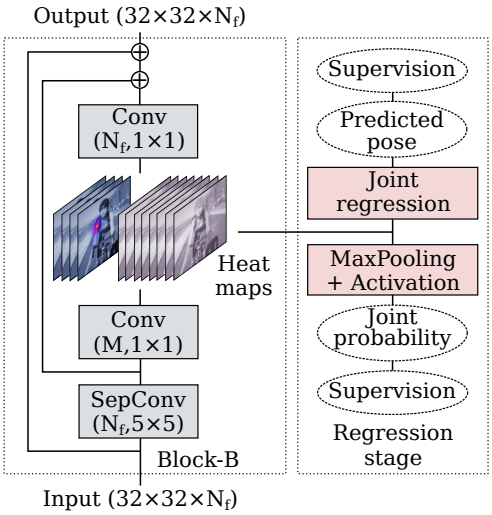


Figure 4.5 – Network architecture of block-B and an overview of the regression stage. The input is projected into M heat maps ($M_d + M_c$) which are then used for pose regression.

4.3.1.1 Detection and Context Aggregation

Even if the correlation between some joints can be learned in the hidden convolutional layers, the joint regression approach is designed to locate body parts individually, resulting in low flexibility to learn from the context. For example, the same filters that give high response to images of a clean head, also must react positively to a hat or a pair of sunglasses. In order to provide multi-source information to the final prediction, we include in our framework specialized part-based heat maps and context heat maps, which are defined as $\mathbf{H}_d = [\mathbf{h}^d_1, \ldots, \mathbf{h}^d_{N_J}]$ and $\mathbf{H}_c = [\mathbf{h}^c_{1,1}, \ldots, \mathbf{h}^c_{N_c,N_J}]$, respectively. Additionally, we define the joint probability related to each context map as $\hat{\mathbf{b}}^c_{i,n}$, where $i = \{1, \ldots, N_c\}$ and $n = \{1, \ldots, N_J\}$.

Finally, the n^{th} joint position from detection and contextual information aggregated is given

by:

$$\hat{\mathbf{p}}_n = \alpha \hat{\mathbf{p}}_n^d + (1-\alpha)\frac{\sum_{i=1}^{N_c} \hat{\mathbf{b}}_{i,n}^c \hat{\mathbf{p}}_{i,n}^c}{\sum_{i=1}^{N_c} \hat{\mathbf{b}}_{i,n}^c}, \tag{4.9}$$

where $\hat{\mathbf{p}}_n^d$ = soft-argmax($\mathbf{h}'^d_n$) is the predicted location from the n^{th} part-based heat map, $\hat{\mathbf{p}}_{i,n}^c = soft-argmax(\mathbf{h}'^c_{i,n})$ and $\hat{\mathbf{b}}_{i,n}^c$ are respectively the location and the probability for the i^{th} context map for joint n, and α is a hyper-parameter that controls the ratio between specialized and contextual information.

From Equation 4.9, we can see that the final prediction is a combination of one specialized prediction and N_c contextual predictions pondered by their probabilities. The contextual weighted contribution brings flexibility, allowing specific filters to be more responsive to particular patterns. This aggregation scheme within the learning stage is only possible because we have the joint probability and position directly available inside the network in a differentiable way.

4.3.2 Experiments

We evaluate the proposed method on the very challenging MPII Human Pose [2] and Leeds Sports Poses (LSP) [65] datasets. Some qualitative results of our method, considering indirectly learned heat maps and regressed poses, are shown in Figure 4.6. The details about the datases, the used metrics, the training process and implementation details are given as follows. The results reported are published in [84].

4.3.2.1 Datasets

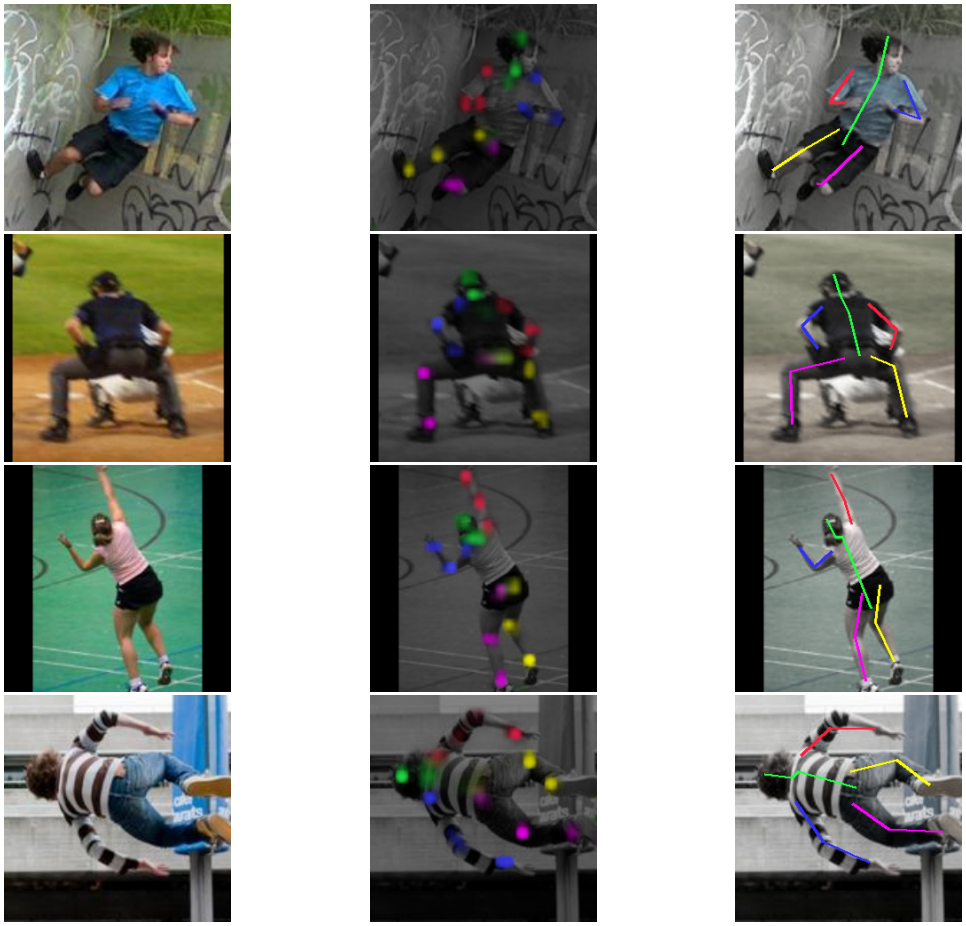
MPII. The MPII Human Pose dataset is composed of about 25K images of which 15K are training samples, 3K are validation samples and 7K are testing samples (which labels are withheld by the authors). The images are taken from YouTube videos covering 410 different human activities, the manually annotated poses have 16 body joints, some of them are not present and others are occluded but can be predicted by the context.

LSP. The Leeds Sports Poses dataset is composed by 2000 annotated poses with up to 14 joint locations. The images were gathered from Flickr with sports people. Two different sets of annotations are provided: Observer-Centric (OC) and Person-Centric (PC).

4.3.2.2 Metrics

For 2D pose estimation, there are three widely used metrics reported in the literature. The first is the Percentage of Correct estimated body Parts (PCP) [39]. In the PCP metric, an estimated joint is considered correct if its distance to the ground truth lie within a fraction of the ground truth segment length. Usually, this fraction is 0.5. The second metric is the Percentage of Correct Keypoints (PCK) [120, 161], which considers that a predicted joint is correct if its distance to the ground truth is smaller or equal to a fraction of the torso length, usually used as 0.2. The third metric, propose by [3], is similar to the PCK metric, but uses the head segment instead of the torso length as the reference. This metric is called PCKh in allusion to the head reference, and the threshold usually used to report results is 0.5. Specifically, the PCK metric for the n^{th} body joint is given by:

$$\text{PCK}_n(r) = \frac{1}{N_s}\sum_{i=1}^{N_s} \mathbf{1}\left(\frac{\|\hat{\mathbf{p}}_n^i - \mathbf{p}_n^i\|_2}{\|\mathbf{p}_{lhip}^i - \mathbf{p}_{rsho}^i\|_2} \leq r\right), \tag{4.10}$$



(a) Input image (b) Part-based maps (c) Regressed poses

Figure 4.6 – Test samples from the Leeds Sports Poses (LSP) dataset. Input image (a), the predicted part-based maps encoded as RGB image for visualizasion (b), and the regressed pose (c). Corresponding human limbs have the same colors in all images. This figure is better seen in color.

where N_s is the number of samples in the evaluation dataset and r is the threshold fraction. For PCKh, the reference length from the *left hip* to the *right shoulder* is replaced by the head size.

4.3.2.3 Implementation Details

The soft-argmax layer can be easily implemented in recent frameworks by concatenating a spatial softmax layer followed by one non-trainable convolutional layer with 2 filters of size H × W, with fixed parameters according to Equation 4.4.

The network model was defined according to Figure 4.3 and composed of eight prediction blocks (K = 8). We trained the network to regress 16 body joints with 2 context maps for each joint ($N_J = 16$, $N_c = 2$). In the aggregation stage, we use $\alpha = 0.8$.

4.3.2.4 Training

The proposed network was trained simultaneously on pose regression and joint probabilities. For pose regression, we use the elastic net loss function (L1 + L2) [176]:

$$\mathbf{L}_p = \frac{1}{N_J} \sum_{n=1}^{N_J} \|\mathbf{p}_n - \hat{\mathbf{p}}_n\|_1 + \|\mathbf{p}_n - \hat{\mathbf{p}}_n\|_2^2, \tag{4.11}$$

where $\mathbf{p}_n$ and $\hat{\mathbf{p}}_n$ are respectively the ground truth and the predicted n^{th} joint coordinates. In this case, we use directly the joint coordinates normalized to 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 joint probability estimation, we use the binary cross entropy loss function on the estimated joint probabilities $\hat{\mathbf{b}}$:

$$\mathbf{L}_b = \frac{1}{N_J} \sum_{n=1}^{N_J} [(\mathbf{b}_n - 1)\ log\ (1 - \hat{\mathbf{b}}_n) - \mathbf{b}_n\ log\ \hat{\mathbf{b}}_n], \tag{4.12}$$

where $\mathbf{b}_n$ and $\hat{\mathbf{b}}_n$ are respectively the ground truth and the predicted joint probabilities.

We optimize the network using back propagation and the RMSProp optimizer, with batch size of 16 samples. For the MPII dataset, we train the network for 120 epochs. The learning rate begins at 10^{-3} and decreases by a factor of 0.4 when accuracy on validation plateaus. We use the same validation split as proposed in [137]. On the LSP dataset, we start from the model trained on MPII and fine-tuned it for more 70 epochs, beginning with a learning rate of $2 \cdot 10^{-5}$ and using the same decrease procedure. The full training of our network takes three days on the relatively outdated NVIDIA GPU Tesla K20 with 5GB of memory.

We used standard data augmentation on both MPII and LSP datasets. Input RGB images were cropped and centered on the main subject with a squared bounding box, keeping the people scale (when provided), then resized to 256 × 256 pixels. We perform random rotations (±40°) and random rescaling from 0.7 to 1.3 on MPII and from 0.85 to 1.25 on LSP to make the model more robust to image variations.

4.3.2.5 Results

LSP dataset. We evaluate our method on the LSP dataset using two metrics, the "Percentage of Correct Parts" (PCP) and the "Probability of Correct Keypoint" (PCK) measures, as well as two different evaluation protocols, "Observer-Centric" (OC) and "Person-Centric" (PC), resulting in four different evaluation settings. Our results compared to the state-of-the-art on the LSP dataset using

Table 4.1 – Results on LSP test samples using the PCK measure at 0.2 with OC annotations.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Avg. PCK
Detection based methods								
Chu et al. [28]	93.7	87.2	78.2	73.8	88.2	83.0	80.9	83.6
Pishchulin et al. [105]	**97.4**	92.0	83.8	79.0	93.1	88.3	83.7	88.2
Regression based method								
Our method	**97.4**	**93.8**	**86.8**	**82.3**	**93.7**	**90.9**	**88.3**	**90.5**

Table 4.2 – Results on LSP test samples using the PCP measure with OC annotations.

Method	Torso	Upper leg	Lower leg	Upper arm	Fore-arm	Head	PCP
Detection based methods							
Chu et al. [28]	95.4	87.6	83.2	76.9	65.2	89.6	81.1
Pishchulin et al. [105]	96.0	91.0	83.5	82.8	71.8	**96.2**	85.0
Regression based method							
Our method	**98.2**	**93.8**	**89.8**	**85.8**	**75.5**	96.0	**88.4**

OC annotations are presented in Table 4.1 (PCK measure) and Table 4.2 (PCP measure). Complete tables including older results from related methods can be consulted in Appendix B. In both cases, we overcome the best scores by a significant margin, specially with respect to the lower leg and the ankles, on which we increase the results of Pishchulin et al. [105] by 6.3% and 4.6%, respectively.

Using the PC annotations on LSP, we achieve the best results among regression based approaches and the second general score, as presented in Table 4.3 and Table 4.4. On the PCK measure, we outperform the results reported by Carreira et al. [16] (CVPR 2016), which is the only regression method reported on this setup, by 18.0%.

MPII dataset. On the MPII dataset, we evaluate our method using the "Single person" challenge [2]. The scores were computed by the providers of the dataset, since the test labels are not publicly available. As shown in Table 4.5, we reach a test score of 91.2%, which is only 0.7% lower then the best result using detection based method, and 4.8% higher than the second score using regression.

Taking into account the competitiveness of the MPII Human Pose challenge[2], our score represents a very significant improvement over regression based approaches and a promising result compared to detection based methods. Moreover, our method requires less computations than the stacked hourglass network from Newell et al. [93] or its extension from Chu et al. [29], since we perform predictions from features at resolution 32×32 instead of 64×64. Due to limited memory resources, we were not able to train these two models in our hardware. Despite that, we reach comparable results with a model that fits in much smaller GPUs.

[2]MPII Leader Board: http://human-pose.mpi-inf.mpg.de

Table 4.3 – Results on LSP test samples using the PCK measure at 0.2 with PC annotations.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	PCK
Detection based methods								
Bulat and Tzimi. [13]	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Chu et al. [29]	**98.1**	**93.7**	**89.3**	**86.9**	**93.4**	**94.0**	**92.5**	**92.6**
Regression based methods								
Carreira et al. [16]	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Our method	**97.5**	**93.3**	**87.6**	**84.6**	**92.8**	**92.0**	**90.0**	**91.1**

Table 4.4 – Results on LSP test samples using the PCP measure with PC annotations.

Method	Torso	Upper leg	Lower leg	Upper arm	Fore-arm	Head	PCP
Detection based methods							
Bulat and Tzimi. [13]	97.7	92.4	89.3	86.7	79.7	95.2	88.9
Chu et al. [29]	**98.4**	**95.0**	**92.8**	**88.5**	**81.2**	95.7	**90.9**
Regression based methods							
Carreira et al. [16]	95.3	81.8	73.3	66.7	51.0	84.4	72.5
Our method	**98.2**	**93.6**	**91.0**	**86.6**	**78.2**	**96.8**	**89.4**

Table 4.5 – Comparison results with state-of-the-art methods on the MPII dataset on testing, using PCKh measure with threshold as 0.5 of the head segment length. Detection based methods are shown on top and regression based methods on bottom.

Method	Head	Shouler	Elbow	Wrist	Hip	Knee	Ankle	Total
Detection based methods								
Newell et al. [93]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Chu et al. [29]	**98.5**	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chou et al. [27]	98.2	**96.8**	92.2	88.0	**91.3**	89.1	84.9	91.8
Chen et al. [22]	98.1	96.5	**92.5**	**88.5**	90.2	**89.6**	**86.0**	**91.9**
Regression based methods								
Rogez et al. [117]	–	–	–	–	–	–	–	74.2
Carreira et al. [16]	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Sun et al. [130]	97.5	94.3	87.0	81.2	86.5	78.5	75.4	86.4
Our method	**98.1**	**96.6**	**92.0**	**87.5**	**90.6**	**88.0**	**82.7**	**91.2**

4.3.3 Discussion

As suggested in section 4.2.2, the proposed soft-argmax function acts as a constraint on the regression approach, driving the network to learn part-based detectors indirectly. This effect provides the flexibility of regression based methods, which can be easily integrated to provide 2D pose estimation to other applications such as 3D pose estimation or action recognition, while preserving the performance of detection based methods. Some examples of part-based maps indirectly learned by our method are show in Figure 4.7. As we can see, the responses are very well localized on the true location of the joints without explicitly requiring so.

Additionally to the part-based maps, the contextual maps give extra information to refine the predicted pose. In some cases, the contextual maps provide strong responses to regions around the joint location. In such cases, the aggregation scheme is able to refine the predicted joint position. On the other hand, if the contextual map response is weak, the context reflects in very few changes on the pose. Some examples of predicted poses and visual contributions from contextual aggregation are shown in Figure 4.8. The contextual maps are able to increase the precision of the predictions by providing complementary information, as we can see for the right elbows of the poses in Figure 4.8.

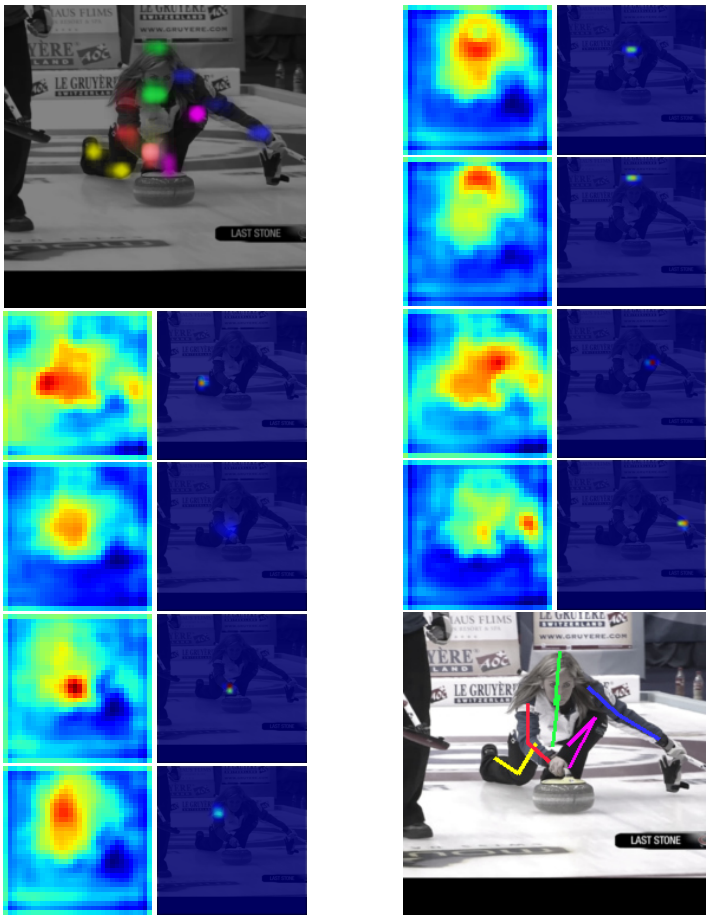
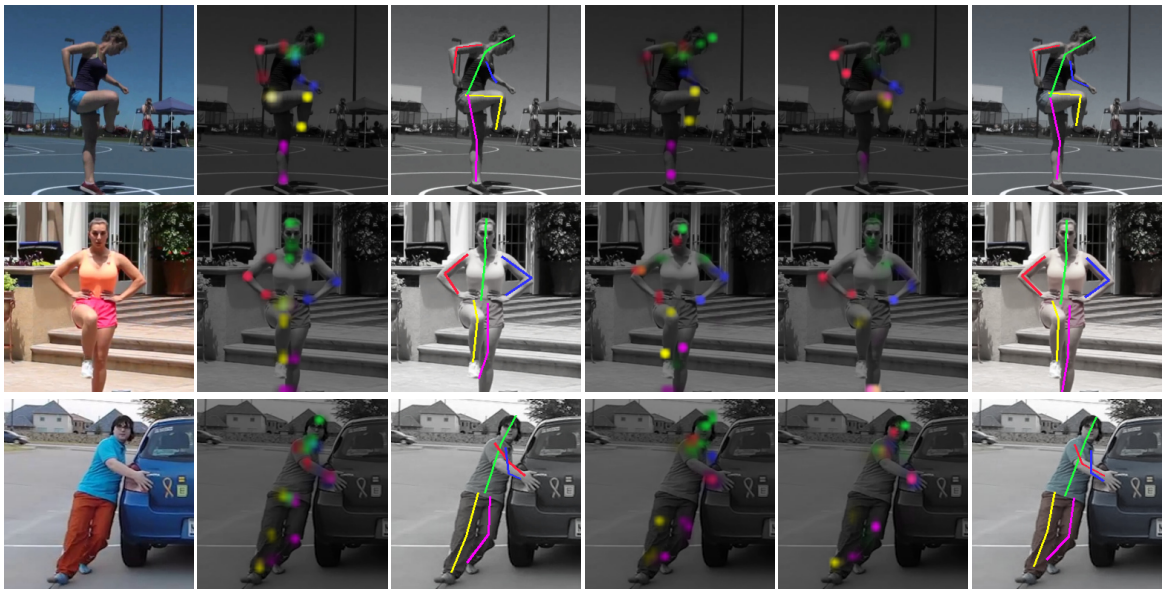


Figure 4.7 – Indirectly learned part-based heat maps from our method. All the joints encoded to RGB are shown in the first image (top-left corner) and the final pose is shown in the last image (bottom-right corner). On each column, the intermediate images correspond to the predicted heat maps before (left) and after (right) the spatial softmax normalization. The presented heat maps correspond to *right ankle*, *right hip*, *right wrist*, *right shoulder*, *upper neck*, *head top*, *left knee*, and *left wrist*.



(a) (b) (c) (d) (e) (f)

Figure 4.8 – Samples of context maps aggregated to refine predicted pose. Input image (a), part-based detection maps (b), predicted pose without context (c), two different context maps (d) and (e), and the final pose with aggregated predictions (f).

4.4 Volumetric Heat Maps for 3D Predictions

In this section, we extend the method introduced in section 4.3 to handle 2D and 3D pose regression in a unified way. The details of our approach are explained as follows.

4.4.1 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 map, and the z component is regressed by applying a one-dimensional soft-argmax on the volumetric representation averaged in both x and y dimensions, as illustrated in Figure 4.9. 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 for action regocnition, as explained in chapter 5.

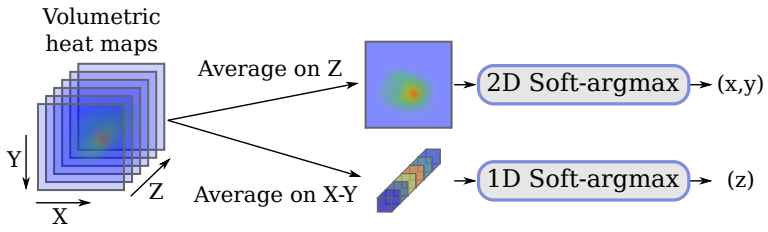


Figure 4.9 – Unified 2D/3D pose estimation by using volumetric heat maps.

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 unconstrained environments, which are usually manually annotated with 2D labels.

4.4.2 Experiments

In this section we present the experimental evaluation of our method considering 2D and 3D human pose estimation. The results are published in [83].

4.4.2.1 Datasets

We evaluate our method on two different datasets: on MPII [3] and on Human3.6M [57] for respectively 2D and 3D pose estimation. The first dataset was previously introduced in section 4.3.2.1. The second one is detailed as follows.

Human3.6M. The Human3.6M dataset [57] is composed by videos with 11 subjects performing 17 different activities and 4 cameras with different points of view, resulting in 3.6M frames. For each person, the dataset provides 32 body joints, from which only 17 are used to compute scores.

4.4.2.2 Metrics

For 2D human pose estimation i.e., for MPII, we use the PCKh metric, as detailed in section 4.3.2.2, and the Area Under the Curve (AUC), varying the reference coefficient from 0.0 until 0.5, with step 0.01. For 3D predictions, we use the standard *mean per joint position error* (MPJPE) metric, which

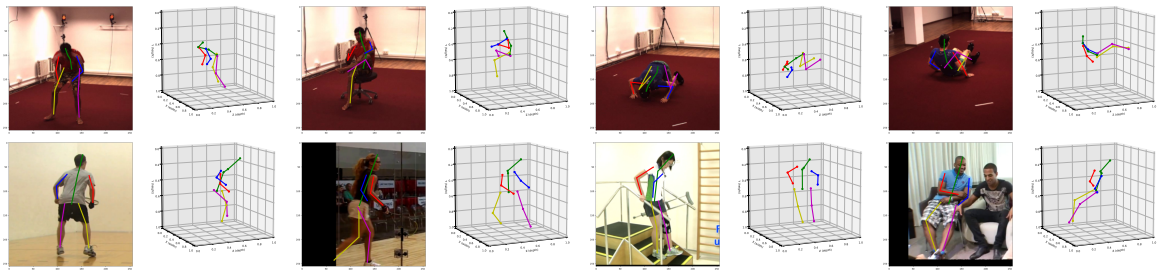


Figure 4.10 – Predicted 3D poses from Human3.6M (top row) and MPII (bottom row) datasets.

measures the average joint error after centering both predictions and ground truth poses to the origin. The MPJPE metric is defined by:

$$\mathrm{E}_{\mathrm{MPJPE}}(\mathbf{p}_o, \hat{\mathbf{p}}_o) = \frac{1}{\mathrm{N}_s} \sum_{i=1}^{\mathrm{N}_s} \|\mathbf{p}_o^i - \hat{\mathbf{p}}_o^i\|_2, \tag{4.13}$$

where N_s is the number of samples, $\mathbf{p}_o$ and $\hat{\mathbf{p}}_o$ are respectively the ground truth and estimated poses after reconstruction and centered in the origin.

4.4.2.3 Implementation Details

We train the network using the elastic net loss function (Equation 4.11) on predicted poses, with the difference that here we do not back propagate the error on z for 2D data. For training, we crop bounding boxes centered on the target person by using the ground truth annotations or the persons location, when applicable. If a given body joint falls outside the cropped bounding box on training, we set the ground truth probability flag to zero, otherwise we set it to one. The ground truth joint visibility information is used to supervise the predicted joint probability vector $\hat{\mathbf{b}}$ with the binary cross entropy loss (Equation 4.12). We optimize the network with 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. We augment the training data by performing random rotations from $-45°$ to $+45°$, scaling from 0.7 to 1.3, and random horizontal flipping.

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, which is not considered in the loss function for this case.

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.

4.4.2.4 Evaluation on 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 4.6. These results are similar to the ones shown in Table 4.5, but here we also present the AUC metric and the PCKh with 0.2 of the head size. 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.

4.4.2.5 Evaluation on 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 [131, 99, 88, 19] 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 4.7 and show that our method is able to outperform the state of the art by a fair margin. Qualitative results are shown in Figure 4.10, 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.

In order to show the contribution of multiple datasets in training, we show in Table 4.8 additional results on 3D pose estimation using Human3.6M only and Human3.6M + MPII datasets for training. When considering multimodal training (mixed data) and single crop, we gain 12.2 mm in precision, which is a very significant improvement for this dataset.

4.4.3 Discussion

In section 4.4.1 we show that the proposed regression method for pose estimation can be easily extended to perform 3D predictions by including one additional dimension in the predicted heat maps. In practice, we extend the idea of contextual maps to learn the additional information related to depth. One of the advantages of this approach is that both 2D and 3D data can be mixed together during training, since we back propagate or not the prediction error, based on the type of input data. This technique allows us to improve precision on 3D predictions substantially, thanks to the multimodal training.

Table 4.6 – 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					
Stacked Hourglass [93]	2016	66.5	33.4	90.9	62.9
Fractal NN [95]	2017	–	–	91.2	63.6
Multi-Context Att. [29]	2017	67.8	**34.1**	91.5	63.8
Self Adversarial [27]	2017	**68.0**	34.0	91.8	63.9
Adversarial PoseNet[22]	2017	–	–	91.9	61.6
Pyramid Res. Module[155]	2017	–	–	**92.0**	**64.2**
Regression methods					
LCR-Net [117]	2017	–	–	74.2	–
Compositional Reg.[131]	2017	–	–	86.4	–
2D Soft-argmax		**67.7**	**34.9**	**91.2**	**63.9**

Table 4.7 – 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 et al. CVPR'17	67.4	71.9	66.7	69.1	71.9	65.0	68.3	83.7
Sun et al. ICCV'17	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7
Ours (single-crop)	**51.5**	**53.4**	**49.0**	**52.5**	**53.9**	**50.3**	54.4	**63.6**
Ours (multi-crop + h.flip)	**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
Pavlakos et al. CVPR'17	96.5	71.4	76.9	65.8	59.1	74.9	63.2	71.9
Sun et al. ICCV'17	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
Ours (single-crop)	**73.5**	**55.3**	**61.9**	**50.1**	**46.0**	**60.2**	**51.0**	**55.1**
Ours (multi-crop + h.flip)	**70.9**	**53.7**	**60.3**	**48.9**	**44.4**	**57.9**	**48.9**	**53.2**

Table 4.8 – 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
Human3.6 only - SC	64.1	66.3	59.4	61.9	64.4	59.6	66.1	78.4
Human3.6 only - MC	61.7	63.5	56.1	60.1	60.0	57.6	64.6	75.1
Human3.6 + MPII - SC	51.5	53.4	49.0	52.5	53.9	50.3	54.4	63.6
Human3.6 + MPII - MC	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
Human3.6 only - SC	102.1	67.4	77.8	59.3	51.5	69.7	60.1	67.3
Human3.6 only - MC	95.4	63.4	73.3	57.0	48.2	66.8	55.1	63.8
Human3.6 + MPII - SC	73.5	55.3	61.9	50.1	46.0	60.2	51.0	55.1
Human3.6 + MPII - MC	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2

4.5 Scalable Sequential Pyramid Networks

In this section, we present a new neural network architecture called Scalable Sequential Pyramid Networks (SSP-Net), which main characteristics are its scalability *a posteriori* i.e., after training, and its dense multi-level supervision with re-injection. We also propose a new 3D pose regression approach, departing from requiring the expensive volumetric heat maps. The details about the proposed network is presented in the following.

4.5.1 Network architecture

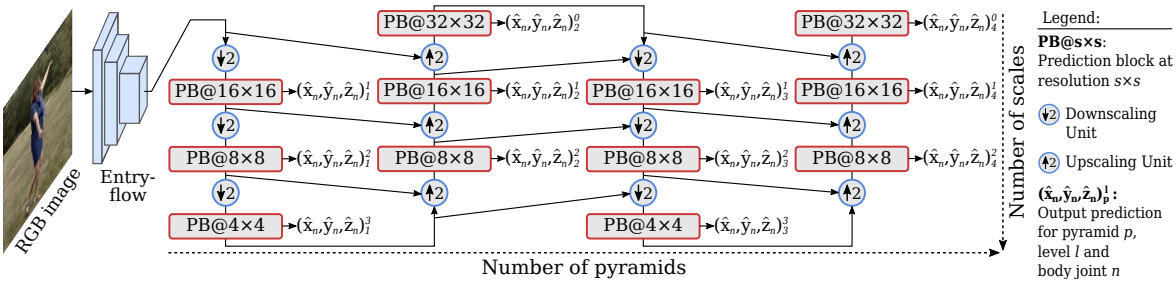


Figure 4.11 – Global architecture of SSP-Net. The entry-flow extracts a preliminary feature map from the input image. These features are then fed through a sequence of CNNs composed of *prediction blocks* (PB) connected by alternating *downscaling* and *upscaling units* (DU and UU). Each PB outputs a supervised pose prediction that is refined by further blocks and units. See Figure 4.12 and Figure 4.13 for the architectural details of DU, UU, and PB.

The global architecture of the proposed network is presented in Figure 4.11, which is a combination of four modules: *entry-flow*, *downscaling* and *upscaling units*, and *prediction blocks*. The role of the entry-flow (detailed in Table 4.9) is to provide deep convolutional features extraction. These features are successively downscaled and upscaled, respectively by downscaling and upscaling pyramids. Each downscaling or upscaling pyramid is composed of a sequence of downscaling or upscaling units (DU or UU), interleaved with prediction blocks (PB) at each level. Prediction blocks are indexed by the pyramid index $p \in \{1,2,\ldots,N_p\}$, where N_p is the number of pyramids, and by the level $l \in \{0,1,\ldots,N_l\}$, where N_l denotes the number of downscaling/upscaling steps performed. Note that in this arrangement, an odd p index corresponds to a downscaling pyramid and an even p index corresponds to an upscaling pyramid.

The architectural details of DU and UU are shown in Figure 4.12. The basic building block

Table 4.9 – Entry-flow network.

	Layer	Filters	Size/strides	Output
	Input	3		256 × 256
	Convolution	64	7 × 7/2	128 × 128
	Convolution	64	1 × 1	
	Convolution	128	3 × 3	
	Residual			128 × 128
	MaxPooling		3 × 3/2	64 × 64
	Convolution	128	1 × 1	
2×	Convolution	256	3 × 3	
	Residual			64 × 64
	MaxPooling		2 × 2/2	32 × 32
	Convolution	192	1 × 1	
2×	Convolution	384	3 × 3	
	Residual			32 × 32

of the pyramid networks is the separable residual block (Figure 5.8a), which consists of a depth wise separable convolution [25] with a residual connection. Our choice for depth wise separable convolutions is mainly due to its benefits in efficiency [50]. One important advantage from our approach is the combination of features from different pyramids and levels. This is performed in both DU/UU, since they combine features from lower/higher levels, as well as features from previous pyramids.

Details of the prediction block (PB) are shown in Fig. 4.13. It takes as input a feature map $\mathscr{X}_p^l$, considering pyramid p and level l, and produces a set of heat maps $\mathbf{h}_p^l$ and depth maps $\mathbf{d}_p^l$, which are used for *3D pose regression*. Heat maps and depth maps generation is defined in the following equations:

$$\mathscr{Y}_p^l = \mathrm{ReLU}(\mathrm{BN}(\mathrm{SC}(\mathscr{X}_p^l))), \tag{4.14}$$

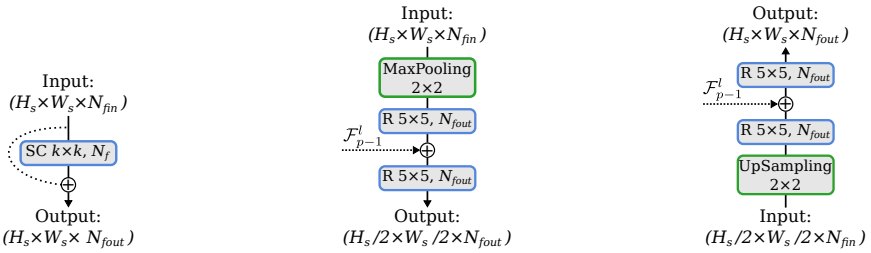
$$\mathbf{h}_p^l = \mathbf{W}_h^{p,l} * \mathscr{Y}_p^l, \tag{4.15}$$

$$\mathbf{d}_p^l = \mathbf{W}_d^{p,l} * \mathscr{Y}_p^l, \tag{4.16}$$

where $\mathscr{Y}_p^l$ is an intermediate feature representation, SC is a separable convolution, $\mathbf{W}_h^{p,l}$ and $\mathbf{W}_d^{p,l}$ are weight matrices with shape $\mathbb{R}^{\mathrm{N}_f \times \mathrm{N}}$, respectively for heat maps and depth maps projection, and $*$ is the convolution operation. Additionally, each prediction block also produces a new feature map $\mathscr{F}_p^l$, which combines the input features with predicted heat maps and depth maps, and is used by next blocks and units for further improvements. This step is defined in equation 4.17:

$$\mathscr{F}_p^l = \mathscr{X}_p^l + \mathscr{Y}_p^l + \mathbf{W}_r^{p,l} * \mathbf{h}_p^l + \mathbf{W}_s^{p,l} * \mathbf{d}_p^l, \tag{4.17}$$

where $\mathbf{W}_r^{p,l}$ and $\mathbf{W}_s^{p,l}$ are called re-injection matrices.



(a) Sep. residual block (b) Downscaling unit (c) Upscaling unit

Figure 4.12 – Elementary blocks of the proposed network. In (a), the separable residual block is used as the basic building block. In (b-c), the downscaling unit (DU) and upscaling unit (UU) take as secondary input the feature maps $\mathscr{F}_{p-1}^l$ issued from the previous piramid. SC: separable convolution; R: separable residual block; $\mathrm{H}_s \times \mathrm{W}_s$: features size; $\mathrm{N}_{fin/fout}$: number of input/output features.

Differently from the Stacked Hourglass [93, 99] architectures, where only the higher resolution features are supervised, we use intermediate supervision at every level of the pyramids. Adding more supervisions does not significantly increase the computational cost of our method, since contrarily to the Stacked Hourglass we do not need to generate artificial ground truth heat maps. On the other hand, with intermediate supervisions in multiple levels we enforce the robustness of our method to variations in the scale of feature maps, while efficiently increasing the receptive field of the global network. Furthermore, our architecture injects the predictions from these intermediate supervisions back into the network by merging them with the current features. This allows the subsequent blocks to perform refining operations instead of full predictions.

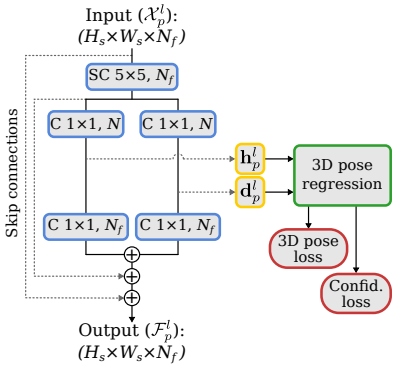


Figure 4.13 – Network architecture of prediction block. Input features $\mathscr{X}_p^l$ (for pyramid p and level l) are used to produce heat maps $\mathbf{h}_p^l$ and depth maps $\mathbf{d}_p^l$, from which 3D pose and confidence scores are estimated. Output features $\mathscr{F}_p^l$ are a combination of input features and re-injected predictions. C: convolution; SC: separable convolution; $\mathrm{H}_s \times \mathrm{W}_s$: features size; N_f: number of features; N: number of body joints.

4.5.2 Joint Based Attention for Depth Estimation

In our approach, we propose to regress 3D joint coordinates from two different mappings: heat maps for (x, y) coordinates and depth maps for z. As a natural choice, we split the problem as 2D regression and depth estimation. For 2D regression, we use the approach already introduced in section 4.2 to recover the (x, y) coordinates. For depth estimation, we propose a new attention mechanism guided by 2D joint estimation. Our method does not require any parameter and is fully differentiable.

For each body joint, we estimate its relative depth $\hat{\mathbf{z}}$ with respect to the root joint, which is usually designated by the pelvis. Specifically, we define an attention mechanism for predicted depth maps based on the appearance information encoded in heat maps. Considering one probability map $\mathbf{h}'$ and the respective depth map $\mathbf{d}$, both with size $\mathbb{R}^{\mathrm{H}\times\mathrm{W}}$, the estimated relative depth is given by:

$$\hat{\mathbf{z}} = \frac{\sum_{i=1}^{\mathrm{H}} \sum_{j=1}^{\mathrm{W}} \mathbf{d}_{i,j} \mathbf{h}'_{i,j}}{\sum_{i=1}^{\mathrm{H}} \sum_{j=1}^{\mathrm{W}} \mathbf{h}'_{i,j}}, \tag{4.18}$$

which can be interpreted as a selection of relevant regions from $\mathbf{d}$ based on the response from $\mathbf{h}'$. In our implementation, values in depth maps are normalized in the interval $[0, 1]$, corresponding to a range of depth prediction, end the probability map $\mathbf{h}'$ is positive and normalized.

The 3D poses estimated by our approach are composed by the (x, y) coordinates in pixels (Equation 4.5) and by the z coordinate relative to the root joint. In order to recover the absolute 3D pose in world coordinates, we require the absolute depth of the root joint and the camera calibration parameters to convert pixels into millimeters. As we show later, estimating the absolute 3D pose directly in world coordinates is not the most relevant problem, since the camera calibration can affect such a prediction drastically. On the other hand, the relative position of joints with respect to the root is of high relevance, and usually is the only measure used to compare different methods. We show in the experiments that absolute depth of the root joint can be estimated without major impact on accuracy.

4.5.3 Experiments

We evaluate the proposed SSP-Net quantitatively on two challenging datasets for 3D human pose estimation: Human3.6M [57] and MPI-INF-3DHP [88]. We also use the manually annotated MPII

Human Pose dataset (2D only) [2] to improve the quality of low level visual features of our network by mixing it with the other two datasets in a 50%/50% ratio on each training batch, as our conclusion of section 4.4. We show in Figure 4.14 some samples of indirected learned heat maps in different pyramid scales and 3D poses estimated by our approach.

4.5.3.1 Datasets

MPI-INF-3DHP. MPI-INF-3DHP [88] is a recent dataset for 3D human pose estimation. It was recorded with a markerless MoCap system, which allows videos to be recorded in outdoor environment e.g., TS5 and TS6 from testing. A total of 8 actors were recorded performing 8 activities sets each. The activities involve some complex exercising poses, which makes this dataset more challenging than Human3.6M.

4.5.3.2 Metrics

For Human3.6M, we use the already introduced MPJPE metric (Equation 4.13) between predicted and ground truth poses. For MPI-INF-3DHP, the authors proposed three evaluation metrics: the mean per joint position error (MPJPE), in millimeters, the 3D Percentage of Correct Keypoints (PCK_{3d}), and the Area Under the Curve (AUC) for different threshold on PCK_{3d}. The PCK_{3d} is similar to the one from Equation 4.10, but the 3D coordinates in millimiters are used and the standard threshold factor is 150mm. Differently from previous works, we use the real 3D poses to compute the error instead of the normalized 3D poses, since the last one cannot be easily computed from the image plane.

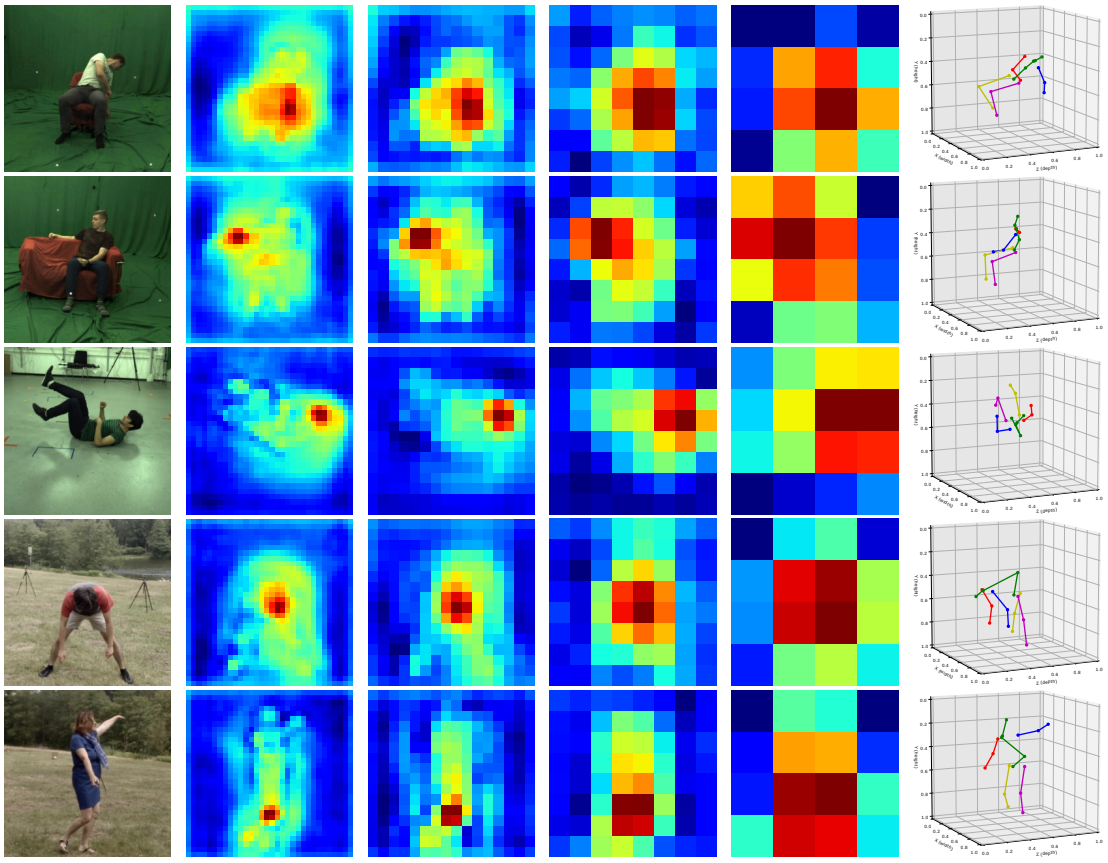
4.5.3.3 Implementation Details

Similarly to the previous sections, we train the network using the elastic net loss function (Equation 4.11) on predicted poses, considering both 2D and 3D data. For the joint confidence scores, we use the binary cross entropy loss where $\mathbf{c}_n$ and $\hat{\mathbf{c}}_n$ are respectively the ground truth and the predicted confidence scores. We use $\mathbf{c}_n = 1$ if the n^{th} joint is present in the image and $\mathbf{c}_n = 0$ otherwise. For the depth (z coordinate), the root joint is assumed to have $z = 0.5$, and a range of 2 meters is used to represent the remaining joints, which means that $z = 0$ corresponds to a depth of -1 meter with respect to the root.

The network architecture used in our experiments is implemented according to Fig. 4.11 and is composed of 8 pyramids, divided as 4 downscaling and 4 upscaling pyramids, each one with 4 scales ($N_p = 8$ and $N_l = 3$). We optimize the network using back propagation and RMSprop with batches of 24 images and initial learning rate of 0.001, which is divided by 10 when validation score plateaus. We used standard data augmentation on all datasets, including: random rotations ($\pm 45°$), random bounding box rescaling with a factor from 0.7 to 1.3, and random brightness gain on color channels from 0.9 to 1.1.

4.5.3.4 Results on 3D Pose Estimation

Human3.6M. Table 4.7 shows our results compared to recent methods, where we achieve **50.2 mm** average MPJPE considering multi-crop and **51.6 mm** single-crop at 120 fps. Our approach achieves results comparable to the state-of-the-art overall, and improves individual activities up to 12.4% on "Photo" and 7.7% on "Sit down", which is the most challenging case. In general, our



(a) (b) (c) (d) (e) (f)

Figure 4.14 – Input image samples (a), and their respective heat maps for selected joints at different pyramid scales (b, c, d, e), and the final predicted 3D pose (f) with all the body joints.

method improves state-of-the-art on individual activities even on single-crop at full speed, running on a desktop GeForce GTX 1080Ti GPU, which is, to the best of our knowledge, better than any previous method. Additionally, with the proposed architecture, our approach can be even faster with a small decrease in performance, as shown in the ablation study.

Table 4.10 – Comparison results with previous work on Human3.6M using the MPJPE (millimeters errors) evaluation on reconstructed poses. To reconstruct poses, we use the absolute z of the root joint. MC: multi-crop, using 5 different bounding boxes with horizontal flip.

Methods	Dir.	Disc.	Eat	Greet	Phone	Posing	Purch.	Sit
Volumetric heat maps	49.2	51.6	**47.6**	50.5	51.8	48.5	51.7	61.5
SSP-Net 120 fps	**46.1**	**50.2**	50.2	**47.5**	**52.0**	**45.9**	**48.5**	**62.3**
SSP-Net +multi-crop	**45.1**	**49.1**	49.0	**46.5**	**50.6**	**44.8**	**47.7**	**60.6**
Methods	SitD.	Smoke	Photo	Wait	Walk	WalkD.	WalkP.	Avg
Volumetric heat maps	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2
SSP-Net 120fps	**66.8**	**53.4**	**54.7**	**45.2**	**41.9**	**54.7**	**45.5**	**51.4**
SSP-Net +multi-crop	**65.4**	**52.0**	**52.8**	**44.2**	**40.6**	**54.1**	**44.4**	**50.2**

MPI-INF-3DHP. Our results on this dataset is presented in Table 4.11. We reached the best average score, with an increasing of 6.6% on PCK_{3d} and reducing the average joint error by more than 20 millimeters, what is a very significant improvement, considering that this dataset involves more realistic activities and has two videos recorded in outdoor environment unseen on training.

Table 4.11 – Comparison results with previous work on MPI-INF-3DHP using the PCK and AUC metrics (higher is better) and the MPJPE metric (lower is better), on reconstructed poses. The absolute z of the root joint was used to reconstruct 3D poses.

Methods	Std. Walk	Exer.	Sit Chair	Croush Reach	OnThe Floor	Sport	Misc.	Avg		
	PCK	PCK	PCK	PCK	PCK	PCK	PCK	PCK	AUC	MPJPE
Zhou et al. [170]	-	-	-	-	-	-	-	69.2	32.5	-
Mehta et al. [88]	86.6	75.3	74.8	73.7	52.2	82.1	77.5	75.7	39.3	117.6
Mehta et al. [90]	**87.7**	77.4	74.7	72.9	51.3	83.3	80.1	76.6	40.4	124.7
Ours	87.1	**85.4**	**85.9**	**81.6**	**68.5**	**88.2**	**83.0**	**83.2**	**44.3**	**96.8**

4.5.3.5 Ablation Study

Here we provide some additional experiments that show the behaviour of our method with respect to the proposed network architecture. In Figure 4.15a, we consider each intermediate supervision of the network as a valid output by cutting the network at that stage, and we show the improvement on accuracy (error decreasing) with respect to the number of pyramids in the network. Additionally, the error with respect to each pyramid scale is also shown. We can clearly see that all the scales are improved by the sequence of pyramids, in such a way that in the last pyramid all scales present very similar error. This evolution can be better seen in Table 4.13, where the error of all intermediate predictions are shown. Note that the precision of our regression method is invariant to the scale of the feature maps, since we reached excellent results with heat maps of 4×4 pixels. The same is not true for detection based approach, like in [99], since the predictions are quantized by the argmax function. The error introduced by this quantization can be observed in Table 4.12, where we compare our regression approach with ground truth volumetric heat maps and argmax.

One important characteristic of our network is that it offers an excellent trade off between performance and speed. In Figure 4.15b we show the per joint error for four pyramids with their respective scales compared to the inference speed. Note that we are able to reach 55.5 millimeters

Table 4.12 – Results on Human3.6M (millimeters error, 2D only), comparing predictions using ground truth heat maps and argmax vs. our regression approach.

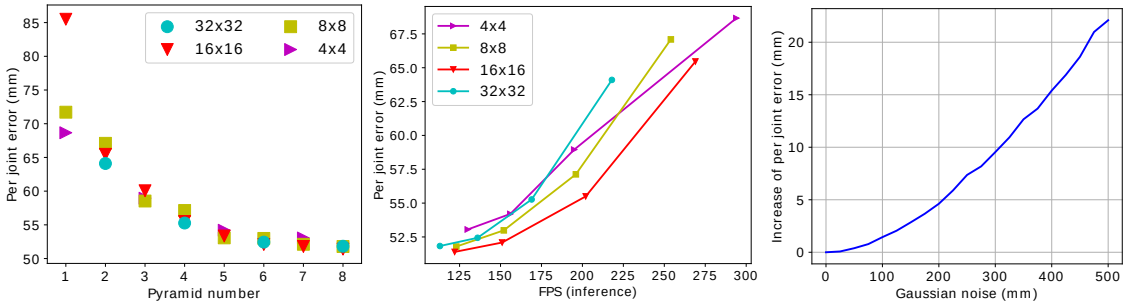
Method / resolution	$s = 4$	$s = 8$	$s = 16$	$s = 32$
Volumetric GT heat maps ($s \times s \times s$) + argmax	233.9	128.6	59.9	31.0
Our regression approach (soft-argmax, Table 3)	53.0	51.8	51.4	51.6

Table 4.13 – Mean per joint position error (MPJPE) in millimeters for all intermediate supervisions of the SSP-Net on the Human3.6M dataset. Odd pyramid numbers correspond to Downscaling Pyramids, and even numbers correspond to Upscaling Pyramids.

Scale	Features res.	Pyramid number / MPJPE							
		1	2	3	4	5	6	7	8
L^0	32×32	-	64.1	-	55.3	-	52.4	-	51.6
L^1	16×16	85.5	65.5	60.1	55.5	55.3	52.1	51.8	51.4
L^2	8×8	71.7	67.1	58.5	57.1	53.1	53.0	52.1	51.8
L^3	4×4	68.7	-	58.9	-	54.2	-	53.0	-

error, which is still a good result on Human3.6M, at a very fast inference rate of 200 FPS.

Finally, we demonstrate on Figure 4.15c the influence of bad prediction of the absolute root depth by adding a Gaussian noise on the ground truth reference. By adding a noise of 100 millimeters (about the same magnitude of the precision of our method on MPI-INF-3DHP), we have an increase in total error inferior to 2 millimeters. This clearly reinforces our idea that the error on relative joint positions is much more relevant than the absolute offset of the root joint.



(a) (b) (c)

Figure 4.15 – Ablation study of our method. In (a), we shown the error performed by each intermediate supervision. The trade off between precision (related to the number of pyramids) and speed is shown in (b), for all the pyramid levels. In (c) we present the increase in reconstruction error with respect to a Gaussian noise injected on absolute root joint position.

4.5.4 Discussion

In this section, we have presented a new neural network architecture able to explore the idea of dense supervisions with re-injection at multiple scales. The method is based on the proposed Scalable Sequential Pyramid Networks, which is a highly scalable network that can be very precise at a small computational cost and extremely fast with a small decrease in accuracy, with a single training procedure. This is possible because the resulting model can be cut after the training procedure. The dense multiscale supervision is possible thanks to our regression approach, which is based on the soft-argmax operation and is invariant to the resolution of feature maps. Addition-

ally, we proposed a new approach to estimate the z coordinate, based on regressed depth maps specialized for each body joint. In this way, we depart from requiring the expensive volumetric heat maps, reducing the network complexity while still reaching state-of-the-art results. We also provided some intuitions about the behaviour of our method in our ablation study, which demonstrates its effectiveness specially for efficient predictions.

4.6 Absolute 3D Human Pose Estimation

In section 4.4 and section 4.5 we presented two different approaches for 3D human pose estimation from RGB images. Differently from the previous methods, where estimated poses are relative to the person center (usually the root joint), in this section we target the problem of predicting body joints in the absolute world coordinates. Despite being a much more challenging task, it has some advantages over relative prediction. For example, absolute prediction eases multiple person separation in scene, and predictions from multiple views can be easily combined in world coordinates, resulting in more precise predictions thanks to the complementarily information from different view-points.

Another problem related to 3D pose estimation is the generalization on unconstrained images, since the majority of 3D datasets have low visual variability due to very constrained acquisition conditions. To circumvent this problem, images "in-the-wild" with 2D annotations or even synthesized images [20, 118] are frequently used as data augmentation. We propose to alleviate this problem by proposing a set of structural constraints on predicted poses, enforcing plausible predictions even for 2D annotated data. Additionally, methods that benefit from multiple datasets for training are restrained to a subset of body joints, which corresponds to the intersection of all datasets. Frequently, even the intersection of labeled body joints is not fully compatible. This limitation could be avoided by using the union among all possible layouts. The downsides are the sparsity of target labels and the increment in complexity, specially for methods relying on costly 3D heat maps for each body joint. We handle this problem by proposing a new skeleton layout, which is the intersection between many popular human pose datasets, resulting in 34 body joints with no ambiguity for joints semantically similar. For each training batch, the additional unlabelled joints are weakly-supervised by the structural constraints.

An overview of the propose method is shown in Figure 4.16. Given an RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, we define the problem of absolute 3D pose estimation as the prediction of $\{\hat{\mathbf{p}}, \hat{\mathbf{c}}, \hat{\mathbf{z}}\}$ from $\mathbf{I}$, where $\hat{\mathbf{p}} \in \mathbb{R}^{N_J \times 3}$ is a predicted pose normalized in the image space and composed of N_J body joints, $\hat{\mathbf{c}} \in \mathbb{R}^{N_J \times 1}$ is an array of confidence scores, one per body joint, $\hat{\mathbf{z}} \in \mathbb{R}^{1 \times 1}$ is a vector with the normalized absolute depth for the root joint, corresponding to the z coordinate orthogonal to the image plane.

Our method can be summarized in a straightforward pipeline, considering two different stages: *training* and *inference*. In the training stage, a pose in real world coordinates (designated by $\mathbf{p}_w$) is projected to the image $\mathbf{I}$, resulting in a pose in image space, composed of pixel coordinates (U-V) and absolute depth for each joint. This projection is designated by $\mathbf{p}_{uvd}$. Then, the pose in the image space is normalized such as its coordinates lie in the interval $[0, 1]$, resulting in $\mathbf{p}$. The regression function f, implemented as a CNN, is trained to predict $\hat{\mathbf{p}}$ and the absolute depth of the root joint from the input image $\mathbf{I}$. At inference time, given an input image, the regression function predicts a normalized pose $\hat{\mathbf{p}} \approx \mathbf{p}$ with its respective joint confidence scores and absolute root joint depth, which are used to recover $\hat{\mathbf{p}}_{uvd}$. Finally, the inverse projection is applied to $\hat{\mathbf{p}}_{uvd}$, resulting in the estimated pose $\hat{\mathbf{p}}_w$ in absolute world coordinates.

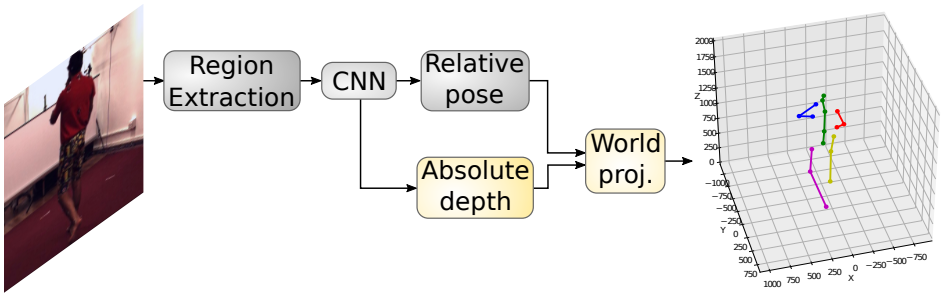


Figure 4.16 – Overview of the proposed method for absolute 3D pose estimation. Given an RGB image, we estimate relative 3D pose centered on the root joint and absolute depth, which are combined to project estimations into the world absolute coordinate.

4.6.1 Absolute Depth Regression

For each predicted pose, an associated value corresponding to the absolute depth with respect to the camera is also estimated. Specifically, given an image patch represented in the feature space by the region Ω, two sources of information are defined: the relative position and size of Ω with respect to the full input image, designated by $\mathscr{P}_\Omega$, and deep convolutional features $\mathscr{F}_\Omega$. The last one is extracted by an average pooling from features with kernel size corresponding to the size of Ω. The relative position is defined as $\mathscr{P}_\Omega = [x_\Omega, y_\Omega, w_\Omega, h_\Omega]^{\mathrm{T}}$, with (x, y) and (w, h) corresponding to the center and size of the image patch with respect to the full frame. Both extracted features are then feed to a fully-connected network with 256 neurons at each level and a single neuron as output. Figure 4.17 illustrates the features extraction process described above and the network architecture.

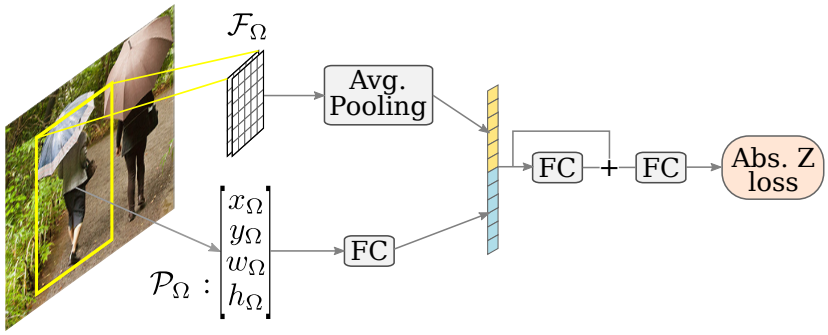


Figure 4.17 – Features and network architecture for absolute depth regression.

4.6.2 Human Pose Layouts

Pose estimation datasets not only have disparate number of body joints, but also the semantically equivalent joints can have deviations from one dataset to another. For example, the joint “head” from Human3.6M is not at the same position as the joint “head” from 2D datasets. Considering that recent works have demonstrated a significant gain in performance by merging different datasets for pose estimation [131, 11], a decision on how to combine different annotations is required. In this work, we decide to use the union of available data by proposing a new Extended Skeleton Template (EST) compose of 34 body joints, as illustrated in Figure 4.18. We show in section 4.6.4.4 results comparing the intersection of body joints, referred as Basic Skeleton Template, with the proposed layout, which evidences the advantage of our approach.

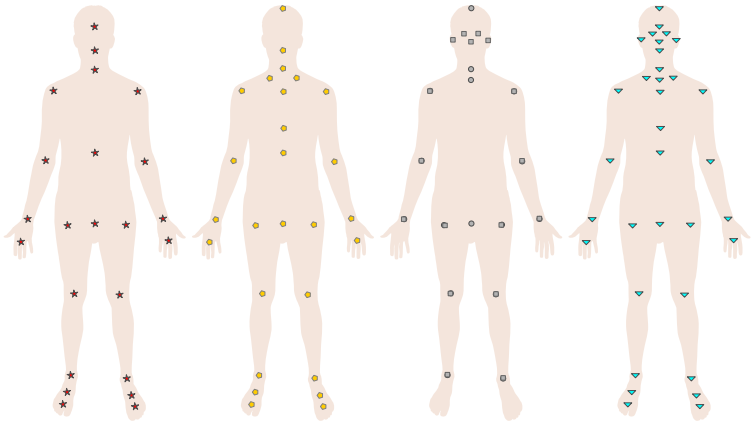


Figure 4.18 – Disposition of body keypoints on pose layouts. From left to right: (a) full layout from Human3.6M, (b) full layout from MPI-INF-3DHP, (c) available keypoint on common 2D datasets, and (d) the proposed Extended Skeleton Template with 34 joints.

4.6.3 Structural Regularization

The downside of having an overdefined skeleton layout is that each training sample has partial ground truth depending on the dataset it comes from. This effect can make specific joints overfit on their own dataset and produce abnormal predictions from other datasets. To mitigate that problem, we include a structural regularization on predicted poses. Since our method relies on a fully-differentiable function, adding a set of constraints on coordinate predictions is a straightforward process. The proposed structural loss is based on length comparisons between individual skeleton segments in the 3D space, resulting in a loss completely invariant to scale, position and rotation.

Considering a predicted pose $\hat{\mathbf{p}} \in \mathbb{R}^{N_J \times 3}$ and its corresponding per joint confidence score $\hat{\mathbf{c}} \in \mathbb{R}^{N_J \times 1}$, we define an elementary structural loss as:

$$\mathscr{L}_s(i, j, \rho) = \hat{\mathbf{c}}_i \hat{\mathbf{c}}_j \big(\| \hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j \|_2 - \rho \big)^2, \tag{4.19}$$

where i and j are indexes of body joints and ρ is a reference length, that could be either another body segment or a reference size normalized to the predicted pose size. By multiplying the squared error between segments by the confidence scores, we ensure that low confidence joints (usually not visible) will not be excessively penalized. The final structural loss is the sum of all elementary losses, which in our method include body symmetry (right-left) and rules of segment reference size. The reference sizes were estimated from millions of available 3D poses from Human3.6M.

4.6.4 Experiments

In this section we demonstrate the effectiveness of our method through a sequence of ablation studies. As detailed next, we evaluate our approach quantitatively on two well known 3D human pose datasets.

4.6.4.1 Datasets

We evaluate the proposed method on Human3.6M [57] and on MPI-INF-3DHP [88], as previously detailed in section 4.5.3.1.

4.6.4.2 Evaluation Protocols and Metrics

For 3D pose estimation on Human3.6M, three evaluation protocols are widely used. In *protocol 1*, six subjects (S1, S5, S6, S7, S8, S9) are used for training and S11 is used for evaluation. Videos for evaluation are sub-sampled every 64th frames, and predictions are aligned to ground truth poses by a Procrustes Alignment before applying the error metric. In *protocol 2*, five subjects (S1, S5, S6, S7, S8) are dedicated for training and S9 and S11 for evaluation. Similarly, evaluation videos are sub-sampled every 64th frames, but *no rigid alignment* is used. The third protocol is the official test set (S2, S3, S4), of which ground truth poses are withheld by the authors and evaluation is performed over all test frames (almost 1 million images) through a server. In our experiments, we consider *protocol 2* for the ablation study in addition to reported results on the test set. *Protocol 1* is not used in this work since the rigid alignment makes the task much easier and therefore less meaningful to evaluate absolute 3D prediction.

The standard metric for Human3.6M is the *mean per joint position error* (MPJPE), which measures the average joint error after centering both predictions and ground truth poses to the origin. Since that loss does now allow to measure the error in absolute world coordinates, we propose a new metric called *mean per joint absolute position error* (MPJAPE), which is computed in world coordinates as:

$$\mathrm{E}_{\mathrm{MPJAPE}}(\mathbf{p}_w, \hat{\mathbf{p}}_w) = \frac{1}{\mathrm{N}_s} \sum_{i=1}^{\mathrm{N}_s} \|\mathbf{p}_w^i - \hat{\mathbf{p}}_w^i\|_2, \tag{4.20}$$

where $\mathbf{p}_w$ and $\hat{\mathbf{p}}_w$ are respectively ground truth and estimated poses.

In the MPI-INF-3DHP dataset, evaluation is performed on a test set composed of 6 videos/subjects, of which 2 are recorded in outdoor scenes, resulting in almost 25K frames. The authors of [88] proposed three evaluation metrics: the mean per joint position error, in millimeters, the 3D Percentage of Correct Keypoints (PCK), and the Area Under the Curve (AUC) for different threshold on PCK. The standard threshold for PCK is 150mm. Differently from previous work, we use the real 3D poses to compute the error instead of the normalized 3D poses, since the last cannot respect a constant camera inverse projection.

4.6.4.3 Implementation Details

For the *backbone network*, we use a pre-trained ResNet cut at *block 4*. To recover features resolution, we use a *head network* composed of one transposed convolution with kernel size 2×2 and strides 2, Followed by a depth-wise convolution with kernel size 3×3. We include a *refinement network* composed of two U-blocks with 4 levels each. Batch normalization and RMSprop are used for training, with starting learning rate of 0.001, decreased by 0.2 after 150K and 170K iterations. Batches of 24 images are used.

The elementary U-block used in the *refinement network* is detailed in Figure 4.19, considering input feature maps of size $32 \times 32 \times 512$. Depthwise residual blocks are similar to standard residual blocks, but use the less costly depthwise convolutions.

During training, the ground truth pose, confidence scores and absolute depth of the root joint are provided for each anchor, normalized based on the anchor size and position. We augmented training data with frequently used techniques, such as random rotations ($\pm 45°$), re-scaling (from 0.7 to 1.3), horizontal flipping, color gains (from 0.9 to 1.1), and artificial occlusions with rectangular black boxes (for indoor datasets only). Additionally, we used popular 2D pose datasets for augmenting training data, which follows our conclusions from section 4.4.

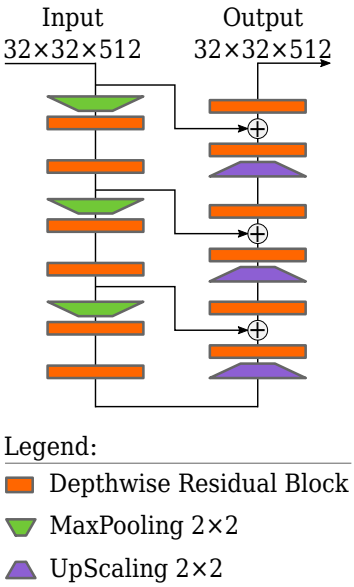


Figure 4.19 – Elementary architecture of U-blocks used in the refinement network.

Table 4.14 – Evaluation of the network architecture, considering the backbone only (ResNet) and the refinement network (ResU-Net), for two skeleton layouts. The number of trainable parameters and prediction error (Human3.6M validation set) are given for comparison.

Pose layout	ResNet		ResU-Net	
	MPJPE	Param.	MPJPE	Param.
BST_{17j}	62.2	10.5M	53.0	23.3M
EST_{34j}	61.9	10.5M	51.1	23.4M

4.6.4.4 Ablation Study

Network Architecture and Pose Layout. We conducted experiments with two different network architectures and two pose layouts, as shown in Table 4.14. An off-the-shelf ResNet pre-trained on Imagenet with the head network (referred simply as ResNet) performed 62.2 mm error using the Basic Skeleton Template (BST). When adding the refinement network (referred as ResU-Net), we gain from 9.2 to 10.7 mm, depending on the skeleton layout. We can see in the refinement network an improvement of 1.9 mm just by replacing the skeleton layout by the Extended Skeleton Template (EST). From this, we can see that precision increases by adding complementary joints, which helps to avoid ambiguity and gives additional information despite marginally increasing the number of parameters. All these results were obtained by mixing 3D and 2D data in the same ratio. When not taking into account 2D data and using only 3D training data from Human3.6M, the average error increases from 53.0 to 64.4 mm.

Absolute Depth Estimation As previously detailed, we use two sources of information for the absolute depth estimation (see Fig.4.17). In order to evaluate how important each of these features are, we evaluated the absolute position error considering (i) only pose and size features $\mathscr{P}_\Omega$, (ii) only deep visual features $\mathscr{F}_\Omega$, and (iii) combined features. Results in absolute mm are presented in Table 4.15, and show that both features are highly complementary.

The Effect of Multiple Camera Views

In this part, we evaluate the effect on prediction from multiple camera views. Since our method predicts 3D poses in world coordinates, we can use multiple cameras to predict the same pose at inference time. In that case, we simply average predicted poses in world coordinates from differ-

Table 4.15 – Absolute position error in mm based on different features combinations for the absolute depth estimation.

Features	Position $\mathscr{P}$	Deep features $\mathscr{F}$	Combined
MPJAPE	539	100.1	91.2

Table 4.16 – Comparison with results from related methods on Human3.6M *test set* using MPJPE (millimeters error) evaluation.

Methods	Dir.	Disc.	Eat	Greet	Phone	Posing	Purch.	Sit
Ionescu et al. [57]	152	153	125	171	135	180	162	168
Grinciunaite et al. [44]	91	89	94	102	105	99	112	151
Popa et al. [108]	60	56	68	64	78	67	68	106
Zanfir et al. [165]	54	54	63	59	72	61	68	101
Ours multi-camera	**34**	**44**	**59**	**45**	**64**	**41**	**55**	**83**

Methods	SitD.	Smoke	Photo	Wait	Walk	WalkD.	WalkP.	Avg
Ionescu et al. [57]	221	160	241	176	157	201	187	171
Grinciunaite et al. [44]	239	109	151	106	101	141	106	119
Popa et al. [108]	119	77	85	64	57	78	62	73
Zanfir et al. [165]	109	74	81	62	**55**	75	**60**	69
Ours multi-camera	**104**	**56**	**61**	**40**	83	**66**	67	**60**

ent views in order the get a single world prediction, then we compute the error with respect to the ground truth. Figure 4.20 illustrates an example of improvement based on multiple camera views. This scenario was evaluated on Human3.6M (see the cameras layout in Fig. 4.21) and our results of relative and absolute error are shown in Table 4.17. As we can see, each camera lowers the error by about 5mm, which is significant on Human3.6M.

4.6.4.5 Comparison with the State of the Art

Human3.6M In Table 4.16, we show our results on the test set from Human3.6M, which is also available in the official leader board H36M_NOS10 track[3].

Since recent approaches frequently only release results on *protocol 2* using validation data, we also compared our method in this scenario, as shown in Table 4.18. Our method obtains state-of-the-art results in single camera, and significantly improves these measures in the 4 cameras setup. We believe these results are close to the best of what can be achieved given the precision of the annotations. Note that we also included our results considering absolute world prediction (MPJAPE) at the bottom of Table 4.18, despite all compared methods being unable to make such prediction and reporting results only on relative MPJPE. This sets a very strong first result for this new challenging task.

[3] Human3.6M leader board: http://vision.imar.ro/human3.6m/ranking.php

Table 4.17 – Results of our method on root joint relative and absolute prediction error (MPJPE / MPJAPE) considering single and multi-camera with different combinations.

Method	MPJPE	MPJAPE
Single camera	51.1	91.2
Single camera + h. flip	**49.2**	**89.5**
Cameras 1,2	45.9	78.8
Cameras 1,4	46.6	84.2
Cameras 1,2,3	41.8	62.7
Cameras 1,2,3,4	**36.9**	**54.7**

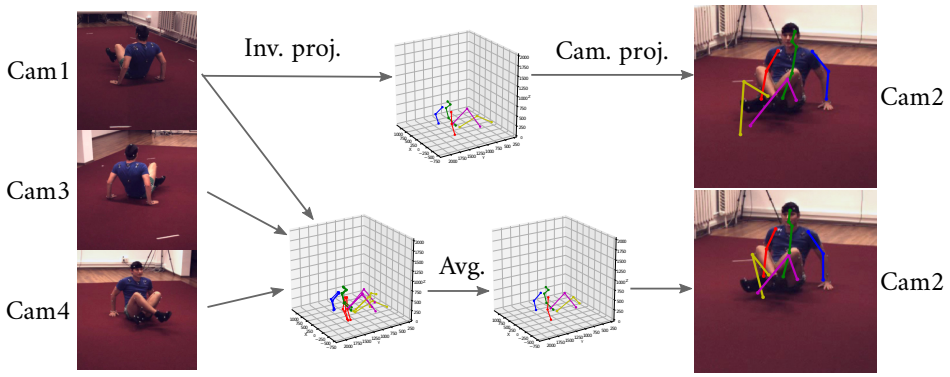


Figure 4.20 – On top, the prediction in world coordinates from camera 1 is projected to camera 2. In the bottom, the averaged predictions from cameras 1, 3, and 4 is projected to camera 2.

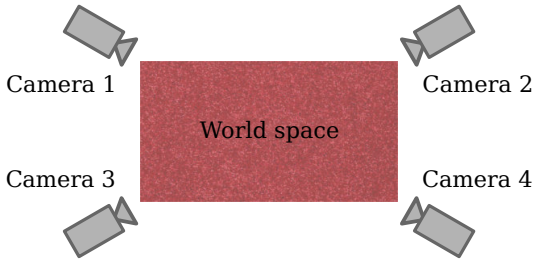


Figure 4.21 – Disposition of cameras on Human3.6M.

MPI-INF-3DHP Our results on MPI-INF-3DHP are shown in Table 4.19. We do not report results considering multiple views in this dataset, since the testing samples were captured by a single camera. As we can see, our method achieves state of the art performances on average and on almost all actions apart from standing.

Table 4.18 – Comparison with results from related methods on Human3.6M *validation set*, protocol 2. We report our scores using two metrics, MPJPE and MPJAPE, on single and multi-camera. Note that all previous methods reported scores *only* on MPJPE.

Methods	Dir.	Disc.	Eat	Greet	Phone	Posing	Purch.	Sit
Sun et al. [131]	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7
Luvizon et al. [83]	49.2	51.6	47.6	50.5	51.8	48.5	51.7	61.5
Sun et al. [132]	–	–	–	–	–	–	–	–
Ours$_{\text{MPJPE}}$ single-camera	**43.3**	**48.3**	**44.9**	**45.2**	**51.5**	**42.7**	**46.0**	62.8
Ours$_{\text{MPJPE}}$ multi-camera	**31.0**	**33.7**	**33.8**	**33.4**	**38.6**	**32.2**	**36.3**	**48.2**
Ours$_{\text{MPJAPE}}$ single-camera	**82.8**	**86.7**	**82.4**	**103.0**	**86.2**	**72.4**	**72.0**	**96.8**
Ours$_{\text{MPJAPE}}$ multi-camera	**43.4**	**48.2**	**47.8**	**68.8**	**50.6**	**39.2**	**46.1**	**65.6**

Methods	SitD.	Smoke	Photo	Wait	Walk	WalkD.	WalkP.	Avg
Sun et al. [131]	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
Luvizon et al. [83]	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2
Sun et al. [132]	–	–	–	–	–	–	–	49.6
Ours$_{\text{MPJPE}}$ single-camera	**69.1**	**51.4**	**52.1**	**43.8**	**37.4**	**50.1**	**42.0**	**49.2**
Ours$_{\text{MPJPE}}$ multi-camera	**51.5**	**39.2**	**38.8**	**32.4**	**29.6**	**38.9**	**33.2**	**36.9**
Ours$_{\text{MPJAPE}}$ single-camera	**128.2**	**89.3**	**86.8**	**103.3**	**76.5**	**84.6**	**79.6**	**89.5**
Ours$_{\text{MPJAPE}}$ multi-camera	**96.4**	**53.4**	**51.8**	**68.8**	**40.7**	**51.1**	**44.1**	**54.7**

Table 4.19 – Results on MPI-INF-3DHP compared to the state-of-the-art.

Method	Stand	Exercise	Sit	Crouch	On the Floor	Sports	Misc.	Total		
	PCK	PCK	PCK	PCK	PCK	PCK	PCK	PCK	AUC	MPJPE
Rogez et al. [117]	70.5	56.3	58.5	69.4	39.6	57.7	57.6	59.7	27.6	158.4
Zhou et al. [170]	85.4	71.0	60.7	71.4	37.8	70.9	74.4	69.2	32.5	137.1
Mehta et al. [88]	**86.6**	75.3	74.8	73.7	52.2	82.1	77.5	75.7	39.3	117.6
Ours	83.8	**79.6**	**79.4**	**78.2**	**73.0**	**88.5**	**81.6**	**80.6**	**42.1**	**112.1**

4.7 Conclusion

In this chapter, we study the problem of human pose regression from RGB images. In a first part, we define the soft-argmax operation as an alternative to detection based approach, resulting in a differentiable method for directly 2D joint coordinates regression, easily integrated with CNNs. Additionally, we demonstrate that contextual information can be seamless integrated into our framework by using additional context maps and joint probabilities. The proposed method results in a significant improvement over the state-of-the-art scores from regression methods and very competitive results compared to detection based approaches.

In a second part, we propose to extend the concept of contextual maps to volumetric heat maps, resulting in a unified approach for 2D and 3D pose estimation. Thus, different datasets, containing 3D or only 2D annotations could be used simultaneously for training, boosting the accuracy for 3D pose estimation.

The previous achievements were then combined with a new network architecture and a new 3D regression strategy, where instead of predicting volumetric heat maps, body depth maps are used to encode the third dimension. As a result, we propose a scalable solution, with multi-level intermediate supervisions, capable of producing state-of-the-art 3D pose predictions at 120 fps, or even faster predictions with lower accuracy.

Finally, we propose to estimate 3D human poses in absolute world coordinates instead of root joint centered. Despite being more challenging, this approach allows to combine predictions from multiple views, resulting in a substantial improvement in accuracy. Furthermore, we propose a new pose layout that merges recent 2D and 3D datasets with no ambiguities, combined with a structural regularization that helps the network to predict plausible 3D poses even on unconstrained environments.

Chapter 5

Multitask Framework for Pose Estimation and Action Recognition

Contents

Prologue

Related Articles:

- D. C. Luvizon, D. Picard, H. Tabia. **2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning**. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5137-5146, 2018.

- D. C. Luvizon, D. Picard, H. Tabia. **Multimodal Deep Neural Networks for Pose Estimation and Action Recognition**, *Congrès Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, 2018.

- D. C. Luvizon, D. Picard, H. Tabia. **Multitask Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition**. **pre-print**, 2019.

Context:

Human pose estimation and action recognition are related tasks, since both problems are strongly dependent on the human body understanding. Despite that, most recent methods in the literature handle the two problems separately. In this chapter, we present a multitask framework for human pose estimation and action recognition. To this end, we build on top of the human pose regression approach, previously detailed in chapter 4, by defining two important types of features: pose and appearance. Given a sequence of color images, our method is able to perform action recognition based on extracted pose and appearance features, resulting in a fully trainable pipeline internally constrained by the predicted human body joints. The proposed method benefits from high sharing of parameters and computations between the two tasks by unifying single frame and video clip processing in a single architecture. Additionally, we provide important insights about the challenges related to multitasking by presenting two scenarios: the first case based on sequential learning, i.e., first training only pose estimation and then training only action recognition; and a second scenario considering multitask learning by optimizing a single model to predict both poses and actions simultaneously, which leads to higher accuracy overall. For the full multitask scenario, we also extend the SSP-Net to action recognition, resulting in a scalable network for both pose and action predictions. The proposed method can be trained with data from different categories simultaneously and achieves state-of-the-art results on both tasks.

5.1 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 which could benefit from it. Similarly, human pose estimation has also rapidly progressed with the emergency 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 [24], or action recognition is used as a prior for pose estimation [162, 59].

However, human pose estimation and action recognition are closely related tasks, since both depend on recognizing and understanding humans in all its complexity of movements and their interactions with the world by only observing the visual data, which very often contains aspects not related to the pose or action itself, like clutter background or clothing.

We have shown in chapter 3 that skeleton or pose sequences can be very informative to recognize certain actions like "tennis serve" and "wave hands", for example. However, more subtle and contextual actions become hard to be distinguished only by the pose information, e.g., "use a fan" and "playing with phone", since these actions could be performed very similarly in terms of movements. In these cases, visual clues could be decisive to provide a more reliable decision.

One of the major advantages of deep learning is its capability to perform end-to-end optimization. As suggested by Kokkinos [67], 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 3D pose estimation tasks thanks to the rise of new architectures and the availability of large amounts of data [93, 99]. Similarly, action recognition has recently been improved by using deep neural networks relying on human pose [8] to extract localized features. 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 proposed to solve this problem by extending the differentiable soft-argmax, as previously detailed in chapter 4, 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. The main contributions from this chapter are presented as follows. *First*, we show that human pose estimation and action recognition can be handled by a unique multitask architecture, strengthening the sharing of parameters and computations, and allowing related tasks to benefit one from another. *Second*, we demonstrate that end-to-end optimization results in better action recognition accuracy when compared to separate and sequential pose and action learning. *Third*, the proposed methods can be trained with multimodal data in a seamless way, e.g., 2D images "in-the-wild", 3D highly precise poses, and video clips for action, resulting in robust learned features that benefits the related tasks. *Fourth*, thanks to the human pose estimation stage, which could be seen as an internal constraint, the action recognition task benefits from 3D pose data, even considering that only RGB frames are required as input. Moreover, the predicted poses are also useful to extract localized visual information, which has been proven in the literature as a good practice for action recognition. *Sixth*, the proposed multitask network is scalable without any additional training procedure, which allows us to choose the right trade-off between speed and accuracy *a posteriori*, for both tasks. Finally, we show that the hard problem of multitasking 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 fps, while achieving its best and state-of-the-art scores at 90 fps on a customer GPU.

The remaining of this chapter is organized as follows. In section 5.2, we present a baseline approach by extending the proposed method for 2D/3D human pose estimation to action recognition, considering sequential learning and fine tuning for action recognition only. In section 5.3 we extend the SSP-Net, previously introduced in section 4.5, to action recognition by improving the multitasking aspect with a joint learning procedure. The experimental evaluation of both scenarios, sequential and joint learning, is presented in section 5.4, followed by our conclusions for this chapter in section 5.5.

5.2 Sequential Pose Estimation and Action Recognition

In this section, we extend the previously proposed framework for 2D and 3D human pose estimation from section 4.4 for action recognition. As shown in Figure 5.1, the proposed method can be divided into three parts. The first is a multitask CNN that provides 2D/3D human pose estimations, deep convolutional (visual) features, and body part probability maps. The other two parts are dedicated to action recognition, 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*. Finally, the results of each part are combined to estimate the final action label.

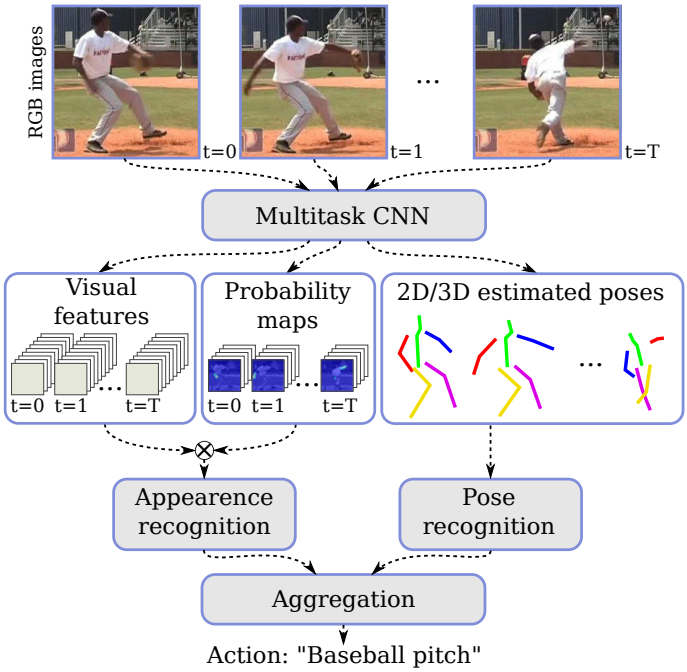


Figure 5.1 – The proposed multitask approach for pose estimation and action recognition based on sequential learning. In this approach we are able to estimate 2D/3D poses from single images and action from frame sequences. Pose and visual information are used to predict actions in a unified framework.

5.2.1 Network Architecture

The multitask CNN is similar to the architecture presented in Figure 4.3. Briefly, it has its entry flow based on Inception-V4 [133], which provides basic features extraction. Eight prediction

blocks are used for pose estimation, and 3D poses are predicted based on volumetric heat maps regression (see section 4.4). 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 5.2.3. A global representation of the pose regression network and the visual features extraction is shown in Figure 5.2.

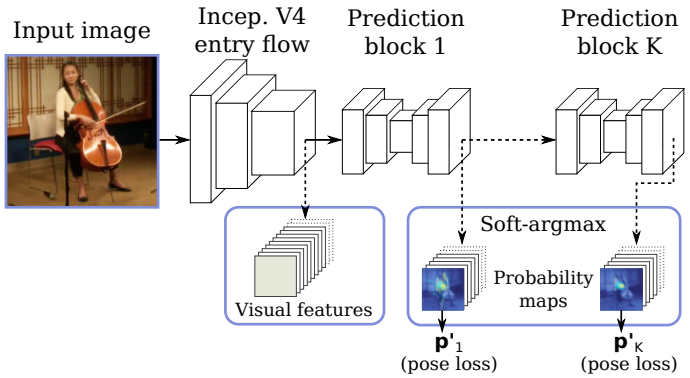


Figure 5.2 – Human pose regression approach from a single RGB frame, used as a base model for action recognition. The input image is fed through a CNN composed by one entry flow and K prediction blocks. Predictions are refined at each prediction block.

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 us 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. In the following, we give a detailed explanation about each action recognition branch, as well as how we extend single frame pose estimation to extract temporal information from a sequence of frames.

5.2.2 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. A similar scheme was proposed by Baradel et al. [8] in a parallel work with ours. 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. The predicted coordinates of each body joints are pondered by their confidence score, 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 Figure 5.3.

We propose a fully convolutional neural network to extract features from input poses and to produce *action heat maps*, as illustrated in Figure 5.4. 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,

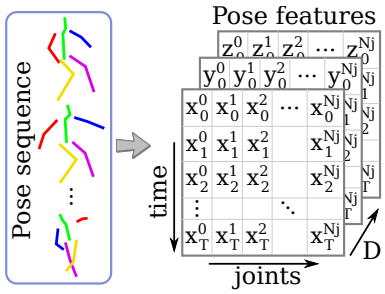


Figure 5.3 – Disposition of pose features for action recognition. Differently from [8], we encode the three dimensions as the channels.

different joints have very different coordinates variations and a filter matching hand patterns will not respond to foot 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 [36] 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.

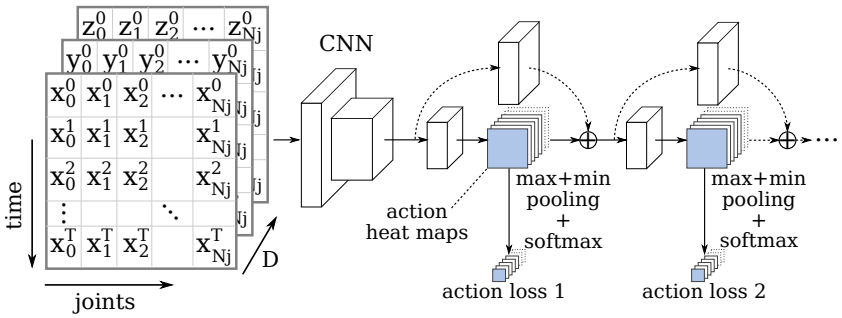


Figure 5.4 – Representation of the architecture for action recognition from a sequence of T frames of N_J body joints. The z coordinates are used for 3D action recognition only. The same architecture is used for appearance-based recognition, except that the input are the appearance features instead of body joints.

5.2.3 Appearance-based Recognition

The extraction of appearance features is 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 multitask features $\mathcal{Z}_t^{p,l} \in \mathbb{R}^{H_f \times W_f \times N_f}$ by each channel from the probability maps $\mathbf{h'}_t \in \mathbb{R}^{H_f \times W_f \times N_j}$, 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. 5.5.

The appearance features are fed into an action recognition network similar to the pose-based action recognition block presented on Figure 5.4 with visual features replacing the coordinates of the body joints. They are similarly arranged in a 2D array stacked vertically for time and horizon-

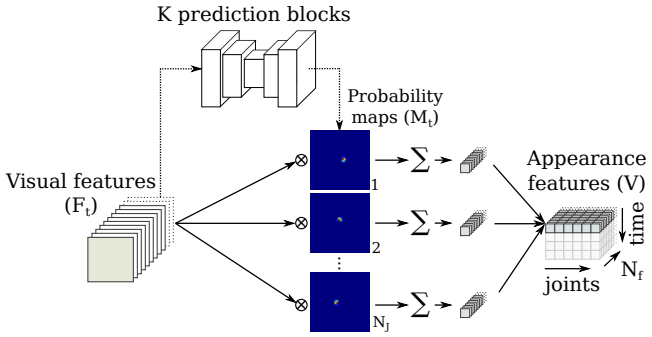


Figure 5.5 – Appearance features extraction from low level visual features and body parts probability maps for a single frame. For a sequence of T frames, the appearance features are stacked vertically producing a tensor where each line corresponds to one input frame.

tally for the body joints .

We argue that our multitask 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.

5.2.4 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. Despite being a simple aggregation strategy, it allows an easy evaluation of each branch’s contribution to the final action prediction.

5.3 Joint Learing Human Poses and Actions

In this section, we present a scalable network architecture for joint human pose estimation and action recognition based on the SSP-Net for pose estimation (from section 4.5). Differently from the previous section, where first we perform pose estimation then action recognition, here we present a new network architecture where pose and action are predicted (and supervised) at different feature map resolutions. Each prediction is re-injected into the network for further refinement. The improvements proposed in this section allow pose estimation and action recognition to be performed in a parallel structure, strengthening the multitask aspect of the network. An overview of the method is shown in Figure 5.6.

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 × W is the frame size. 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

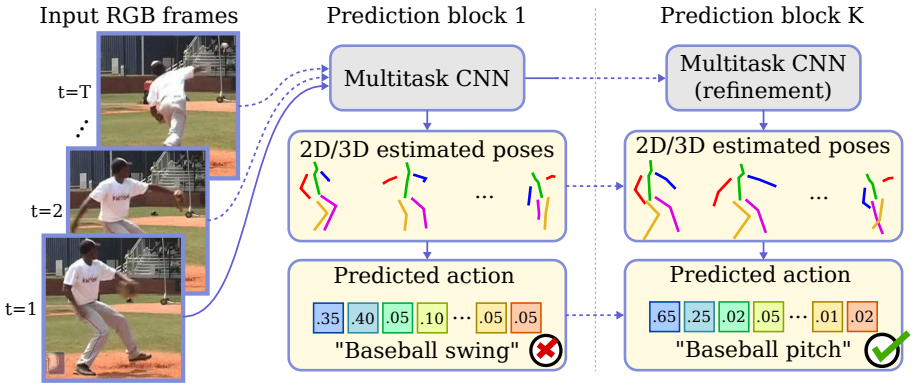


Figure 5.6 – The proposed multitask approach for joint human pose estimation and action recognition. Our method provides 2D/3D pose estimation from single images or frame sequences. Pose and visual information are used to predict actions in a unified framework. Pose and action predictions are refined by K prediction blocks.

$\hat{\mathbf{a}} \in \mathbb{R}^{N_a \times 1}$, 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.

5.3.1 Network Architecture

The global architecture of the proposed method is presented in Figure 5.7. 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 [25] with skip connection. These units are detailed in Figure 5.8. Note that, differently from section 4.5, here both DU and UU have only one convolution. The missing convolution was moved to the prediction block to simplify the visualization of the global architecture. In practice, the architecture for pose estimation remains simillar to the previously introduced SSP-Net.

The network can operate in two distinct modes: (i) *single frame* processing or (ii) *video clip* processing. In the first operational mode, only layers related to pose estimation are active, from which connections correspond to the blue arrows in Figure 5.7. In the second operational mode, 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 on 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 Figure 5.7 respectively by blue and red arrows.

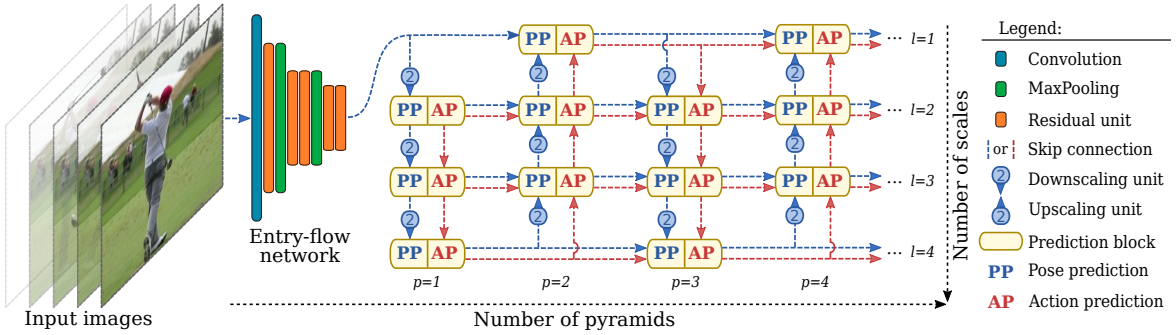


Figure 5.7 – Overview of the proposed multitask network architecture. The entry-flow extracts feature maps from the input images, which are fed through a sequence of CNNs composed of prediction blocks (PB), downscaling and upscaling units (DU and UU), and simple (skip) connections. Each PB outputs supervised 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 Figure 5.8 and Figure 5.9 for details about DU, UU, and PB.

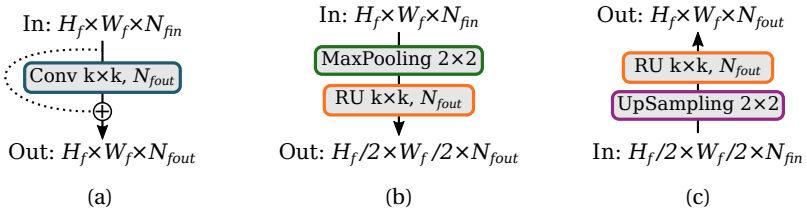


Figure 5.8 – Network elementary units: in (a) residual unit (RU), in (b) downscaling unit (DU), and in (c) upscaling unit (UU). N_{fin} and N_{fout} represent the input and output number of features, $H_f \times W_f$ is the feature map size, and k is the filter size.

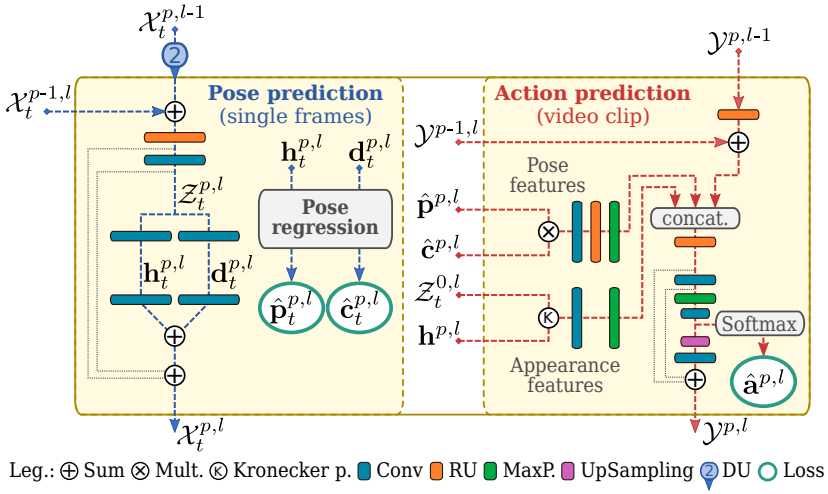


Figure 5.9 – Network architecture of prediction blocks (PB) for a downscaling pyramid. With the exception of the PB in the first pyramid, all PB get as input features from the previous pyramid in the same level ($\mathscr{X}_t^{p-1,l}$, $\mathscr{Y}^{p-1,l}$), and features from lower or higher levels ($\mathscr{X}_t^{p,l\mp1}$, $\mathscr{Y}^{p,l\mp1}$), depending if it composes a downscaling or an upscaling pyramid, respectively.

5.3.1.1 Multitask Prediction Block

The architecture of the prediction block is detailed in Fig. 5.9. 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:

$$\mathscr{X}_t^{p,l} \in \mathbb{R}^{\mathrm{H}_f \times \mathrm{W}_f \times \mathrm{N}_f} \tag{5.1}$$

$$\mathscr{Z}_t^{p,l} \in \mathbb{R}^{\mathrm{H}_f \times \mathrm{W}_f \times \mathrm{N}_f} \tag{5.2}$$

$$\mathscr{Y}^{p,l} \in \mathbb{R}^{\mathrm{T} \times \mathrm{N}_j \times \mathrm{N}_v}. \tag{5.3}$$

Namely, $\mathscr{X}_t^{p,l}$ is a tensor of single frame features, which is propagated from one PB to another, $\mathscr{Z}_t^{p,l}$ is a tensor of multitask (single frame) features, used for both pose and action, $\mathscr{Y}^{p,l}$ is a tensor of video clip features, exclusively used for action predictions and also propagated from one PB to another, $t = \{1,\ldots,\mathrm{T}\}$ 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 $\mathscr{X}_t^{p-1,l}$ from the previous pyramid and the features $\mathscr{X}_t^{p,l\mp1}$ from lower or higher levels, respectively for downscaling and upscaling pyramids. A similar propagation of previous features ($\mathscr{Y}^{p-1,l}$ and $\mathscr{Y}^{p,l\mp1}$) happens for action. Note that both $\mathscr{X}_t^{p,l}$ and $\mathscr{Y}^{p,l}$ feature maps are three-dimensional tensors (2D maps plus channels) that can be easily handled by 2D convolutions.

The tensor of multitask features is defined by:

$$\mathscr{Z}_t^{'p,l} = \mathrm{RU}(\mathscr{X}_t^{p-1,l} + \mathrm{DU}(\mathscr{X}_t^{p,l-1})) \tag{5.4}$$

$$\mathscr{Z}_t^{p,l} = \mathbf{W}_z^{p,l} * \mathscr{Z}_t^{'p,l}, \tag{5.5}$$

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. Then, $\mathscr{Z}_t^{p,l}$ is used to produce body joint probability maps:

$$\mathbf{h}_t'^{p,l} = \Phi(\mathbf{W}_h^{p,l} * \mathscr{Z}_t^{p,l}), \tag{5.6}$$

and body joint depth maps:

$$\mathbf{d}_t^{p,l} = Sigmoid(\mathbf{W}_d^{p,l} * \mathscr{Z}_t^{p,l}), \tag{5.7}$$

where Φ is the spatial softmax [84], and $\mathbf{W}_h^{p,l}$ and $\mathbf{W}_d^{p,l}$ are weight matrices. Probability 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 $\mathbb{R}^{\mathrm{H}_f \times \mathrm{W}_f \times \mathrm{N}_j}$.

5.3.2 Action Features Aggregation and Re-injection

Similarly to the previously discussed approach for sequential learning (section 5.2), in this joint learning version we use the same pose and appearance features as described in subsection 5.2.2 and subsection 5.2.3, respectively. The main difference from the previous version is in the aggregation method. In this approach, a latter aggregation scheme would result in a double flow of action features in the global architecture, one for pose and another for appearance features. To avoid this, we perform an early aggregation of both types of features by concatenating them (jointly with previous action features), as illustrated in the *action prediction* part from Figure 5.9. We observed

slightly better results on action recognition when using early features aggregation.

Similarly to the pose features re-injection mechanism for single frames, as previously discussed in section 4.5, our approach also allows action features re-injection, as illustrated in the action prediction part in Figure 5.9. We demonstrate in the experiments (section 5.4) that this technique also improves action recognition results with no additional parameters.

5.3.3 Decoupled Action Poses

Since the multitask architecture is trained simultaneously on pose estimation and on action recognition, we can have an effect of competing gradients from pose and action, specially in the predicted poses, which is used as the output for the first task and as the input for the second task. To mitigate that influence, we propose to decouple estimated poses (used to compute pose scores) from action poses (used by the action recognition part) as illustrated in Figure 5.10.

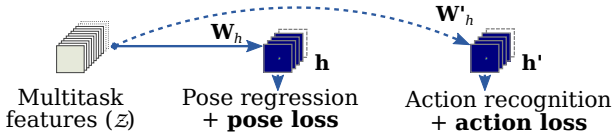


Figure 5.10 – Decoupled poses for action prediction. The weight matrix $\mathbf{W}'_h$ is initialized with a copy of $\mathbf{W}_h$ after an initialization on pose estimation only. The same process is applied to depth maps ($\mathbf{W}_d$ and $\mathbf{d}$).

Specifically, we first train the single frame processing pipeline on pose estimation for a few epochs, then we replicate only the last layers that project the multitask 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. Finally, the video clip processing pipeline for action recognition is based on the replicated poses and continue the training procedure of the full network. 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 epochs), we show in our experiments that they still remain coherent with supervised estimated poses.

5.4 Experiments

In this section, we present quantitative and qualitative results by evaluating the two proposed methods on two different tasks and on two different modalities: human pose estimation and human action recognition on 2D and 3D scenarios. The results here presented are partially published in [83]. We report results on four publicly available datasets, detailed as follows.

5.4.1 Datasets and Evaluation Metrics

For 2D human pose estimation, we evaluate the proposed methods on the MPII human Pose Dataset [2], as previously detailed in section 4.3.2.1. For this task, we use the Percentage of Correct Keypoints based on the head size (PCKh). For 3D pose estimation, we use the Human3.6M dataset [57], as introduced in section 4.4.2.1, and the mean per joint position error (MPJPE) metric

on reconstructed 3D poses. For human action recognition, we report results considering 2D and 3D scenarios, respectively on Penn Action and on NTU RGB+D, which are detailed as follows.

Penn Action. The Penn Action dataset [168] is composed by 2,326 videos with sports people performing 15 different actions, among those "baseball pitch", "bench press", "strum guitar", etc. The challenge on this dataset is due to several missing body parts in many actions and to the disparate image scales from one sample to another.

NTU RGB+D. The NTU dataset [123] 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.

For action recognition, we report results using the percentage of correct action classification score. We use the standard evaluation protocol for Penn Action [154], splitting the data as 50%/50% for training/testing, and the most challenging and 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 one that maximizes the product of the prediction probabilities for all video clips. For cropping bounding boxes, we use our method for pose estimation considering the full image frame, then the region around the estimated pose is expanded by 50% on width and height, resulting in the estimated person bounding box.

In our experimental evaluation we consider two scenarios. First, we evaluate a sequential learning process based on the architecture explained in section 5.2. In the second scenario, we evaluate the joint learning process based on the method described in section 5.3. Both scenarios are detailed respectively in section 5.4.2 and section 5.4.3.

5.4.2 Evaluation on Sequential Learning

5.4.2.1 Training Details

For the sequential learning experiments, we first train the human pose estimation part as previously explained in section 4.4.2. Then, we fixed all the layers corresponding to pose estimation and trained the stacked action recognition part until validation score plateaus. Finally, we train the full network, which we call the *fine tuning* process, for a few more epochs.

For the action recognition task, we train the network using the categorical cross entropy loss, defined as:

$$\mathbf{L}_a = -\sum_{i=1}^{\mathrm{N}_a} \mathbf{a}_i \, log(\hat{\mathbf{a}}_i), \tag{5.8}$$

where $\mathbf{a}$ and $\hat{\mathbf{a}}$ are respectively the one-hot ground truth vector and the predicted action probabilities for one sample. For training, we randomly select fixed-size clips with T frames from a training video sample.

5.4.2.2 2D Action Recognition

We evaluate the proposed 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 is trained using video clips composed of T = 16 frames. Considering the related literature until the date of our publication [83], our method

achieves state-of-the-art results for action classification among methods using RGB and estimated poses. Results are shown in Table 5.1. We reduced the prediction error from 4.7% [14] to 2.6% when using predicted poses, which shows the effectiveness of our method when optimized for both pose and action estimations. We also evaluated our method not considering the influence of estimated poses by using the manually annotated body joints. In this case, our method improves the state of the art by 0.5%, which indicates that the proposed CNN and pose/appearance features are appropriate for the action recognition task.

Table 5.1 – Comparison results on Penn Action for 2D action recognition. Results are given as the percentage of correctly classified actions.

Methods	Annot. poses	RGB	Optical Flow	Estimated poses	Acc.
Iqbal et al. [59]	-	-	-	✓	79.0
	-	✓	✓	✓	92.9
Cao et al. [14]	✓	✓	-	-	98.1
	-	✓	-	✓	95.3
Ours	✓	✓	-	-	**98.6**
	-	✓	-	✓⋆	**97.4**

⋆ Using mixed data from PennAction and MPII.

Table 5.2 – 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
Liu et al. [79]	✓	-	-	74.4
Shahroudy et al. [124]	✓	✓	-	74.9
Baradel et al. [8]	✓	-	-	77.1
	⋆	✓	-	75.6
	✓	✓	-	84.8
Ours	-	✓	-	**84.6**
	-	✓	✓	**85.5**

⋆ GT poses were used on test to select visual features.

5.4.2.3 3D Action Recognition

For 3D action recognition, we consider the pose datasets MPII and Human3.6M, and the action dataset NTU. 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 using only RGB frames and 3D predicted poses, as reported in Table 5.2. If we consider only RGB frames as input, our method improves over [8] 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 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.

5.4.2.4 Ablation Study

We performed varied experiments on NTU to show the contributions of each component of our method. As can be seen on Table 5.3, our estimated poses increase the accuracy by 2.9% over

Table 5.3 – 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

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 5.1) 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 5.11. 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 5.12, the proposed approach also benefits from complementary appearance and pose information which lead to better classification accuracy once aggregated.

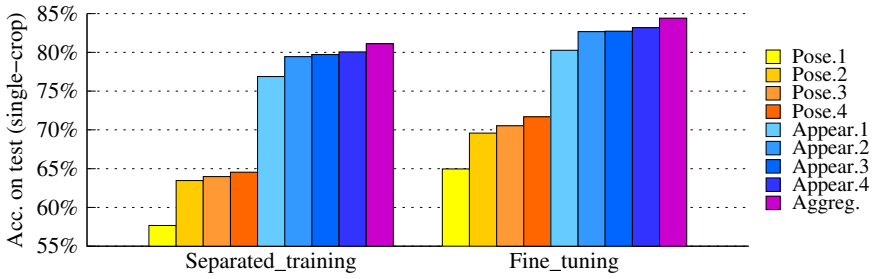


Figure 5.11 – Action recognition accuracy on NTU from pose and appearance models in four prediction blocks, and with aggregated features, for both separated training and full network optimization (fine tuning).

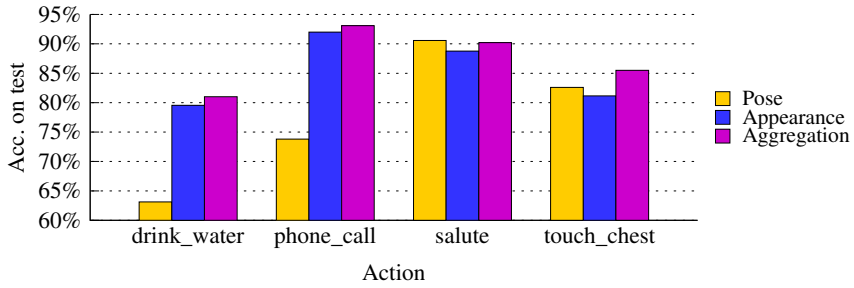


Figure 5.12 – Action recognition accuracy on NTU for different action types from pose, and appearance models and with aggregated results.

5.4.3 Evaluation on Joint Learning

In this section, we evaluate the second proposed method (joint learning), using the same datasets and metrics as in the sequential learning scenario. The differences are mostly related to the archi-

tecture and multitask training details, explained as follows.

5.4.3.1 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 then 8 pyramids. On action recognition, that 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 2D and 3D scenarios, respectively.

5.4.3.2 Multitask 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 multitask scenario, simultaneously for pose estimation and action recognition, until the validation scores plateaus. 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 multitask 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 more 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 multitask training, we use RMSprop for Penn Action with learning rate reduced by a factor of 0.1 after 15 and 25 epochs, and a vanilla SGD for NTU 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 crossentropy loss are much stronger then the gradients from the elastic net loss on pose estimation. Each iteration is performed on four 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 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 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.

5.4.3.3 Evaluation on 3D Pose Estimation

Our results compared to previous approaches are shown in Table 5.4. Our method achieved the state-of-the-art average prediction error of 48.6 millimeters on Human3.6M for 3D pose estimation, improving our baseline (sequential learning with volumetric heat maps) by 4.6 mm. Considering only the pose estimation task, our average error is 49.5 mm, 0.9 mm higher than the multitasking result, which shows the benefit of multitask 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. [158]) by 21 mm. The generalization of our method is demonstrated by qualitative results of 3D pose estimation for all datasets in Figure 5.13. Note that a single model and a single training procedure was used to produce all those images and scores, including 3D pose estimation and 3D action recognition, discussed as follows.

Table 5.4 – 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
Sun et al. [131]	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7
Yang et al. [158]	51.5	58.9	50.4	57.0	62.1	49.8	52.7	69.2
Sun et al. [132]	–	–	–	–	–	–	–	–
3D heat maps (ours [83])	49.2	51.6	47.6	50.5	51.8	48.5	51.7	61.5
Ours (pose only)	**43.7**	**48.8**	**45.6**	**46.2**	**49.3**	**43.5**	**46.0**	**56.8**
Ours (multitask)	**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
Sun et al. [131]	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
Yang et al. [158]	85.2	57.4	65.4	58.4	60.1	43.6	47.7	58.6
Sun et al. [132]	–	–	–	–	–	–	–	49.6
3D heat maps (ours [83])	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2
Ours (pose only)	**67.8**	**50.5**	**57.9**	**43.4**	**40.5**	**53.2**	**45.6**	**49.5**
Ours (multitask)	**64.2**	**50.6**	**53.8**	**44.2**	**40.0**	**51.1**	**44.0**	**48.6**

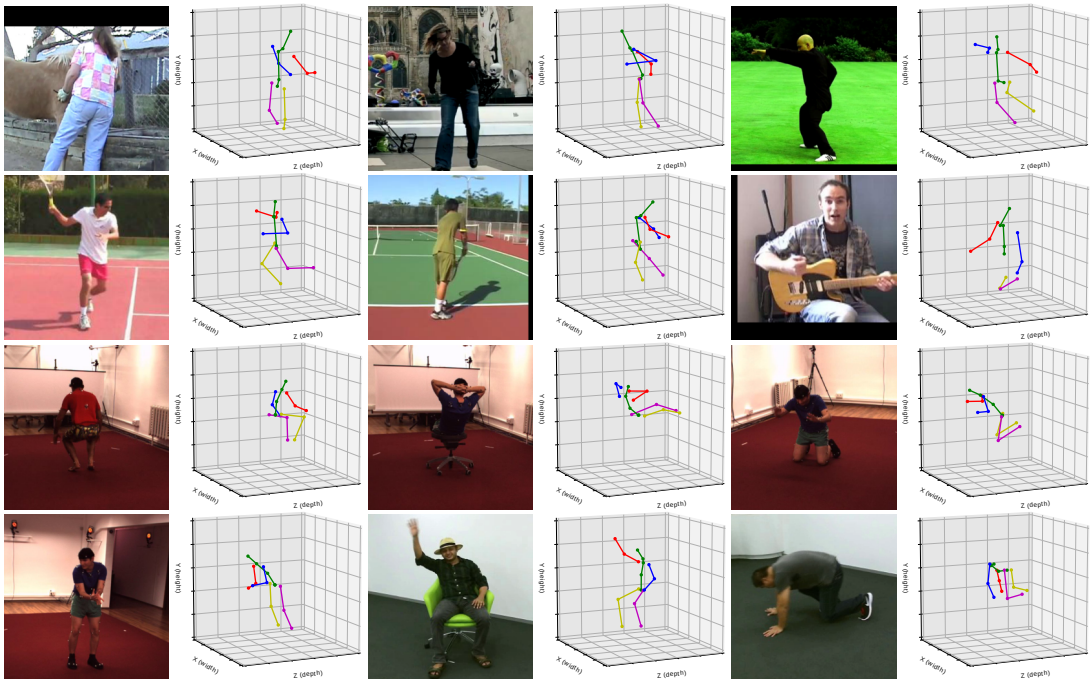


Figure 5.13 – Predicted 3D poses from RGB images for both 2D and 3D datasets.

5.4.3.4 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 5.5. We reached the state-of-the-art action classification score of 98.7% on Penn Action, improving our baseline by 1.3%. Our method outperformed all previous methods, including the ones using ground truth (manually annotated) poses.

Table 5.5 – Results for action recognition on Penn Action. Results are given as the percentage of correctly classified actions.

Methods	Annot. poses	RGB	Optical Flow	Estimated poses	Acc.
Cao et al. [14]	✓	✓	-	-	98.1
	-	✓	-	✓	95.3
Du et al. [35]⋆	-	✓	✓	✓	97.4
Liu et al. [80]†	✓	✓	-	-	98.2
	-	✓	-	✓	91.4
Baseline (ours [83])	✓	✓	-	-	98.6
	-	✓	-	✓	97.4
Ours (single-clip)	-	✓	-	✓	**98.2**
Ours (multi-clip)	-	✓	-	✓	**98.7**

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

For 3D, we trained our multitask 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 5.6. Our approach reached 89.3% of correctly classified actions on NTU, which is a promising result considering the hard task of classifying among 60 different actions in the cross-subject split. Our method improves previous results by at least 2.7% and our baseline by 3.8%, which shows the effectiveness of the proposed joint learning approach.

Table 5.6 – Comparison results on NTU cross-subject for 3D action recognition. Results given as the percentage of correctly classified actions.

Methods	Kinect poses	RGB	Estimated poses	Acc. cross subject
Liu et al. [80]	-	✓	✓	78.8
Baradel et al. [8]	✓	-	-	77.1
	⋆	✓	-	75.6
	✓	✓	-	84.8
Baradel et al. [9]	-	✓	-	86.6
Baseline (ours [83])	-	✓	✓	85.5
Ours	-	✓	✓	**89.3**

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

5.4.4 Ablation Study on Pose and Action Joint Learning

5.4.4.1 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 5.7, 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 performing predictions only on 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. Finally, the last line shows results with the full network, reaching 88.3% on MPII and 98.2% on Penn Action, with a single multitask model.

Table 5.7 – 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 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**

5.4.4.2 Pose and Appearance Features

The proposed method benefits from both pose and appearance features, which are complementary to the action recognition task. Additionally, the confidence scores $\hat{\mathbf{c}}$ are also complementary to pose itself and lead to marginal action recognition gains if used to weight pose predictions, as shown in Figure 5.9. In Table 5.8, we present results on pose estimation and on action recognition for different feature 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 aditional improvements on both tasks. 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 5.8 – 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**

Additionally, we can observe that decoupled action poses remain coherent with supervised poses, as shown in Figure 5.14, which suggests that the initial pose supervision is a good initial-

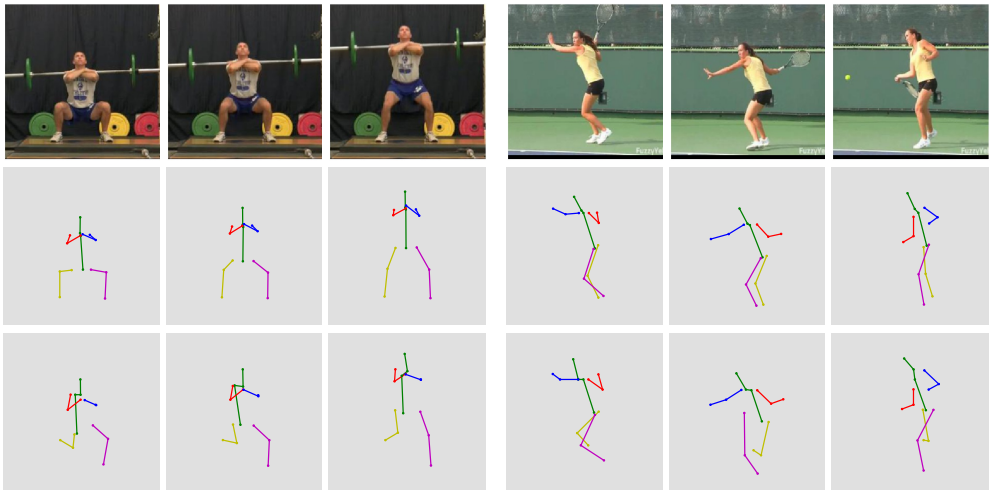


Figure 5.14 – Two sequences of RGB images (top), predicted supervised poses (middle), and decoupled action poses (bottom).

ization overall. Nonetheless, in some cases, decoupled probability maps can drift to regions in the image more relevant for action recognition, as illustrated in Figure 5.15. For example, feet heat maps can drift to objects in the hands, since the last is more informative with respect to the performed action.

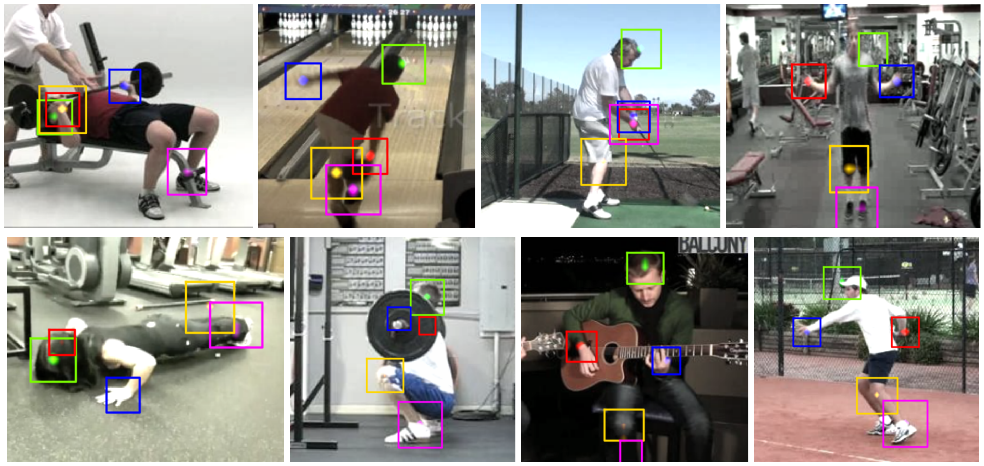
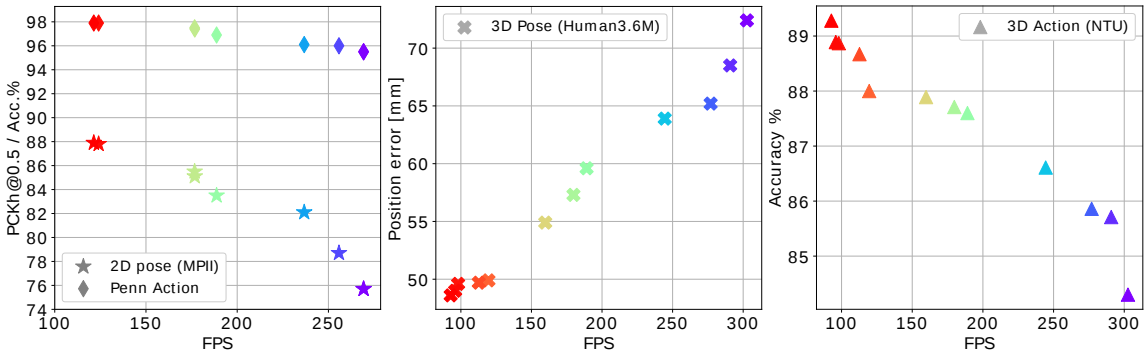


Figure 5.15 – Drift of decoupled probability maps from their original positions (head, hands and feet) used as an attention mechanism for appearance features extraction. Bounding boxes are draw here only to highlight the regions with high responses.

5.4.4.3 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 multitask model at different prediction blocks and estimate the speed inference in frames per second (FPS), evaluating pose estimation precision and action recognition classification accuracy. We consider mini batches with 16 images. The results are shown in Figure 5.16. 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 NTU, at the same speed, we still



(a) (b) (c)

Figure 5.16 – Inference speed of the proposed method considering 2D (a) and 3D (b,c) scenarios. A single multitask model was trained for each scenario. The trained models were cut *a posteriori* for inference analysis.

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 5.9, 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 5.9 – Results on all tasks with the proposed multitask model compared to recent approached 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	Year	MPII PCKh	H36M MPJPE	PennAction *half/half*	NTU RGB+D Cross-sub.
Pavlakos et al. [99]	2017	-	71.9	-	-
Yang et al. [158]	2018	**88.6**	58.6	-	-
Cao et al. [14]	2017	-	-	95.3	-
Du et al. [35]	2017	-	-	97.4	-
Shahroudy et al. [124]	2017	-	-	-	74.9
Baradel et al. [9]	2018	-	-	-	86.6
Ours 2D @ 240 fps	2019	85.5	-	**97.5**	-
Ours 2D @ 120 fps	2019	88.3	-	**98.7**	-
Ours 3D @ 240 fps	2019	80.7	63.9	-	**86.6**
Ours 3D @ 180 fps	2019	83.8	**57.3**	-	**87.7**
Ours 3D @ 90 fps	2019	87.0	**48.6**	-	**89.3**

5.5 Conclusion

In this chapter, we have presented two multitasks deep architectures to perform 2D and 3D pose estimation jointly with action recognition. In the first part, we have presented a sequential learning scheme by first training the pose estimation network, then the action recognition part, and finally fine tuning the full network, from input image sequences to action predictions. The model predicts the 2D and 3D location of body joints from 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 pooling based on human body parts. We also have proposed an action recognition architecture based on multiple prediction blocks, allowing intermediate supervision and prediction re-injection, similarly to common architectures dedicated for pose estima-

tion, resulting in successive incremental improvements. We have demonstrated that fine tuning the model leads to significant improvements on action, for both pose and appearance features.

In the second part, we have presented a joint learning approach by extending the SSP-Net architecture to video clip processing. This architecture benefits from multiple scale processing and multiple scale supervisions, for both pose estimation and action recognition. In this extended architecture, we also initialize pose and action parts individually, but after the initialization stage, the full network is trained simultaneously for pose and action prediction. Similarly, action recognition is performed based on body joint locations and on appearance features extracted in the region of body parts. Despite handling two related but distinct tasks at the same time and with a single model, in this approach we have improved the previous state-of-the-art results significantly on both tasks, while running at more than 90 FPS. Another aspect of the presented multitask approach is its flexibility for on demand requirements for speed. With a single training procedure, our approach offers a wide range of precision vs inference speed.

For highly semantic tasks such as action recognition, it is difficult to converge to an optimal solution with limited training data and without any structural constraints, specially because the input video clips frequently have an enormous amount of data not related to the target action, both on space and time dimensions. By constraining the visual features extraction using the body parts position, the input space is reduced significantly, facilitating the learning process. Moreover, by learning deep visual features with two objectives, pose and action, these features tend to be more robust to visual variations and to encode better representations. This is specially true for the second part of this section, on with single frame and video datasets can be used simultaneously. In this way, not only 3D pose estimation benefits from "in-the-wild" images, as per our conclusions in section 4.4, but also action recognition.

Chapter 6

Conclusion and Perspectives

In this thesis, we have addressed the problems of human action recognition and human pose estimation, considering the use of 3D information. The thesis is divided into three main parts. Each part and its contributions are discussed in the following.

6.1 Main Contributions

6.1.1 Human Action Recognition from Skeleton Sequences

In the first part, we have presented a shallow framework for human action recognition from skeleton sequences estimated from depth maps. In this framework, we proposed to extract local features by forming groups of body parts, composed of 3 to 5 body joints each. Position and motion features are extracted and aggregated using a pool of clusters and the Vector of Locally Aggregated Descriptors (VLAD), resulting in multiple global features that encode the full action sequence. The global features are then combined by a metric learning stage based on the Large Margin Nearest Neighbor (LMNN) algorithm with a structural regularization to reduce overfitting. The learned representations are finally used for action recognition by means of a k-NN classifier. The proposed method achieved state-of-the-art results on three well know skeleton datasets.

The key aspects for the effectiveness of the proposed framework rely on three important points. First, the proposed localized position and motion features encode relevant and complementarily information for action recognition. Second, the randomness associated to the clustering scheme used in the local features aggregation can be effectively reduced by multiple clustering representations, which also offer complementarily information to the global features. Third, the information encoded in the final feature vector can be compressed into a much smaller and more discriminant representation by a metric learning method, even considering that the departing features possibly have redundant information due to the multiple clustering. Finally, by designing each part of the framework to best fit the others, the proposed method consistently achieved effective results on skeleton action recognition.

The limitations of the proposed approach are mainly due to the single source of information used for classification i.e., skeleton sequences. As discussed in chapter 2, predicting skeletons from consumer depth sensors is not feasible under certain unconstrained conditions, in addition to the difficulties imposed by such sensors to combine skeletons and the important visual information accessible through RGB images.

6.1.2 Human Pose Estimation from RGB Images

In the second part of this thesis, we have considered the hard problem of human pose estimation from monocular images. This part was mainly motivated by two observations: first, as noted in chapter 3, precise 3D human poses can be very effective to perform action recognition. However, depth sensors are limited in terms of applicability to real and unconstrained problems, and estimating highly precise 3D poses from RGB images is a difficult task. The second observation comes from the fact that most recent methods for human pose estimation from monocular images are based on detection and use the non differentiable argmax to recover joint coordinates. As a consequence, such methods cannot be easily used for action recognition in a fully differentiable way.

The work developed in this part resulted in a series of important conclusions. We have shown that the proposed soft-argmax for pose regression provides results comparable to the state-of-the-art methods on 2D pose estimation while being fully differentiable and not requiring artificial ground truth generation during training. As a by-product, heat map like representations are learned indirectly by the network. Additionally, with the proposed regression approach, we have demonstrated the consistent improvements on 3D pose estimation by performing multimodal training i.e., by training 3D pose estimation with mixed 3D data and "in-the-wild" images with 2D annotated labels. In the proposed approach, this has become a straightforward training process. We have also proposed a carefully designed network architecture to exploit the most from multi-resolution supervisions, as well as a depth estimation method to predict 3D human poses more efficiently. This new architecture departs from requiring costly volumetric heat maps and provides a range of precision vs inference speed with a single training procedure. Finally, we have shown that the proposed approach can be further extended to predict human poses in absolute coordinates, which has several positive consequences. For example, the absolute position could be used to better distinguish multiple people in a 3D scene, and multiple cameras could be explored to perform more reliable and more precise predictions by merging the absolute estimation from cameras with different view-points.

6.1.3 Multitask Framework for Pose Estimation and Action Recognition

In the last part, we have proposed a multitask approach for human pose estimation and action recognition. We have shown that the proposed human pose estimation method based on the soft-argmax can be used as a building block for action recognition, resulting in a fully differentiable pipeline. For the action recognition part, we have defined two sources of information that can be extract from single frame analysis: human pose, in 2D or 3D body joint coordinates, and appearance features, which are deep CNN features extracted at very specific regions of the input frame, guided by the joint coordinates of estimated poses. The two proposed features are complementarily to each other, and result in robust and effective action recognition when correctly combined. We also have demonstrated that the optimization from end-to-end results in further improvements compared to separated training. This is only possible due to the differentiable pose estimation method.

Not surprisingly, another relevant contribution in the proposed method is its capability to be trained with multimodal data in a seamlessly way, similarly to what is observed in the proposed 3D human pose estimation approach. Specifically, for training the action recognition model, we can benefit from both "in-the-wild" images with 2D annotations and very precise 3D data in controlled environments, as well as video clips for action. This allows the shared layers of the network to

learn robust visual features from distinct modalities of data: single frames and video clips, which respectively contain a rich diversity of visual aspects for learning robust features and the important temporal information for action classification.

Finally, we have extended our efficient multi-resolution SSP-Net architecture for pose estimation and action recognition in a multitask framework. Based upon the two premises that (i) multimodal training is beneficial for action recognition and (ii) learning to solve two related tasks can be beneficial to both of them in a multitasking scenario, we carefully designed a deep neural network to address both problems in a joint way. By consequence, thanks to the joint training procedure, the multitask approach outperformed single tasking. This result was not trivially achieved, but it is intuitive that related tasks could benefit one from another in a deep learning scenario.

6.2 Perspectives and Future Work

The future perspectives for this work are well diversified, depending on the target objectives. Three of the most relevant perspectives are discussed in the following.

6.2.1 Pose Estimation in Absolute Coordinates

Predicting the 3D human pose in absolute world coordinates from monocular images is a very hard problem, but at the same time it has several implications in the way that 3D poses could be used. For example, multiple cameras with different view points could be explored to provide self-constrained predictions in the world coordinates, resulting in more robust predictions against occlusions or clutter background. Additionally, predictions in absolute coordinates could help to distinguish different people with respect to their position in the scene, which is ambiguous with person-centered predictions, since all predictions are at the same absolute depth.

However, the absolute 3D pose prediction from monocular images is a ill-defined problem if no assumptions are made. The height of the predicted persons and the camera calibration could be used as a prior to facilitate the problem. In our method presented in section 4.6, we assume only that the camera calibration is known, but our validation was performed on datasets with controlled environment, on which the absolute depth can be satisfactorily estimated based only on the visual aspects and on the coordinates of the cropped bounding box.

We believe that the proposed method for absolute 3D pose estimation could be extended to a multi-view semi-supervised manner, on which predicted poses in world coordinates from different view-points could be enforced to converge, using the error between two predictions from different viewpoints as an additional supervision. In this scenario, the system would rely on multi-view for the additional supervision, and the absolute predictions would still be possible with single-view, despite of the expected lower precision. Additionally, since the proposed 3D pose regression is fully differentiable, even the camera calibration parameters could be learned for each view-point.

6.2.2 Multi-person 3D Pose Estimation

Multi-person 3D pose estimation is still an open problem with few related methods [117, 165, 89], mainly due to the lack of large-scale multi-person 3D datasets, but also because most of the recent methods for 3D pose estimation are person-centric approaches, which become not a practical solution in the case of multiple persons in the image.

The lack of methods capable of performing pose predictions in absolute coordinates is probably one limiting factor for multi-person 3D pose estimation. By addressing the task as absolute 3D pose estimation, the extension to multi-person becomes naturally easier. To solve the ambiguous problem of absolute depth estimation, deep convolutional neural networks could be exploited to estimate the depth information based on the visual content, possibly enforcing both tasks, 3D pose and depth estimation, in a multitask framework. Additionally, despite resulting in state-of-the-art results on single-person, the soft-argmax is not easily extended to the multi-person scenario, mainly because the spatial softmax normalization is mutually exclusive for multiple body joints in the image patch. A more robust strategy, such as region proposal extractors, associated with the soft-argmax could be investigated in this direction.

6.2.3 Multitask Learning for Action Aspects Disentanglement

The proposed multitask learning method for human pose estimation and action recognition leaves room for other related approaches, specially when considering data with different modalities, as have been proposed for single frame and video clip analysis. Disentangling pose and visual aspects related to actions in videos is a promising direction, since it could be useful for more robust action recognition considering high intra-class variations or even for synthetic video clips generation from the disentangled components. For example, given a video clip, one question which could be answered is: what are the fundamental regions in each frame in order to recognize the performed action? This question could be also formulated with respect to the pose, considering that for some actions, few body joints are relevant.

The challenge in such scenario is how to enforce the disentanglement of the many components related to complex actions, while being able to reconstruct the original input information, in the case of synthetic video generation. For approaches based on global features extraction, e.g., methods using 3D convolutions, it would be a strenuous process, since the high level features encode at once the full aspects of pose, movements, foreground, and background.

On the other hand, in the proposed multitask framework, pose and visual information are already disentangled for each frame. In this case, further extraction of pose aspects related and unrelated to action recognition could be performed with less effort, considering that some joints are much more relevant to some actions than the others. Similarly, the appearance features could also be extended to action related and action unrelated, in such a way that both would be required to reconstruct the video clip frames, but only the first would be required to perform action recognition.

Appendix A

Deep Network Architecture Details

A.1 Implementation Details for the Sequential Pose and Action Model

In this section, we present some additional implementation details related to the 3D pose estimation method based on volumetric heat maps, detailed in section 4.4, and to the sequential learning method for pose and action, decribed in section 5.2.

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 A.1, batch normalization and ReLu activation. The architecture of the multitask stem is detailed in Figure A.2. Each pose estimation prediction block is implemented as a multi-resolution CNN, as presented in Figure A.3. We use $N_d = 16$ heat maps for depth predictions. The CNN architecture for action recognition is detailed in Figure A.4.

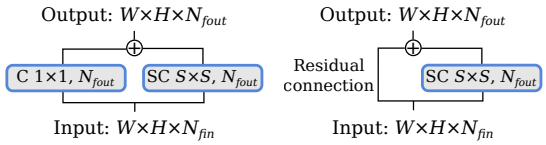


Figure A.1 – Separable residual module (SR) based on depth-wise separable convolutions (SC) for $N_{fin} \neq N_{fout}$ (left), and $N_{fin} = N_{fout}$ (right), where N_{fin} and N_{fout} are the input and output features size, W × H is the feature map resolution, and S × S is the size of the filters, usually 3 × 3 or 5 × 5. C: Simple 2D convolution.

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.

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 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 × 256. We augment the training data by performing random rotations from −45° to +45°, 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.

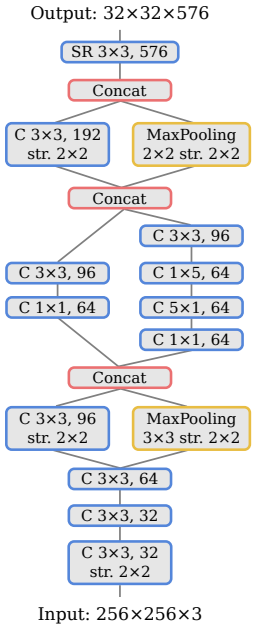


Figure A.2 – Shared network (entry flow) based on Inception-V4. C: Convolution, SR: Separable residual module.

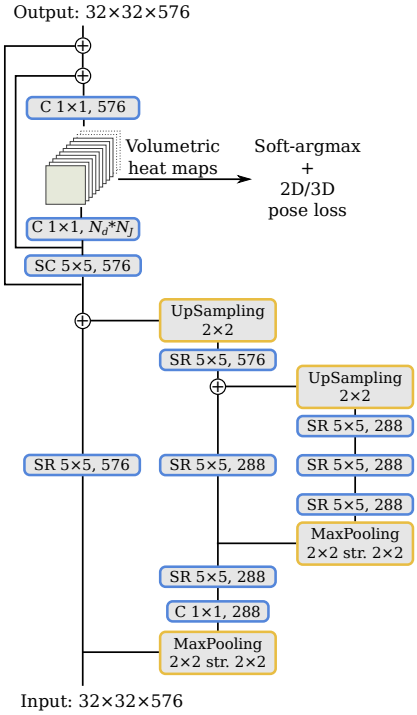


Figure A.3 – Prediction block for pose estimation, where N_d is the number of depth heat maps per joint and N_J is the number of body joints. C: Convolution, SR: Separable residual module.

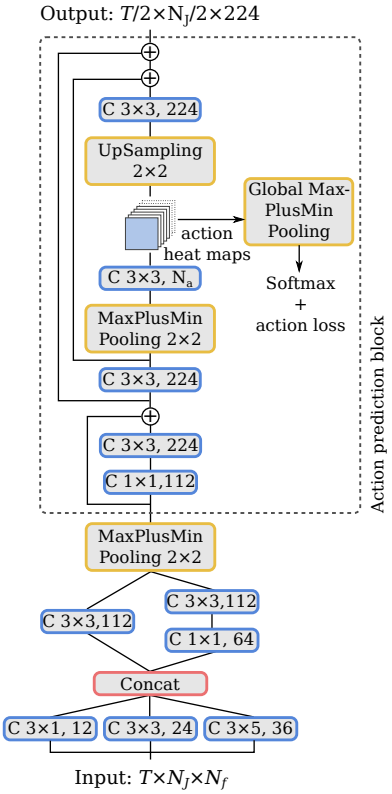


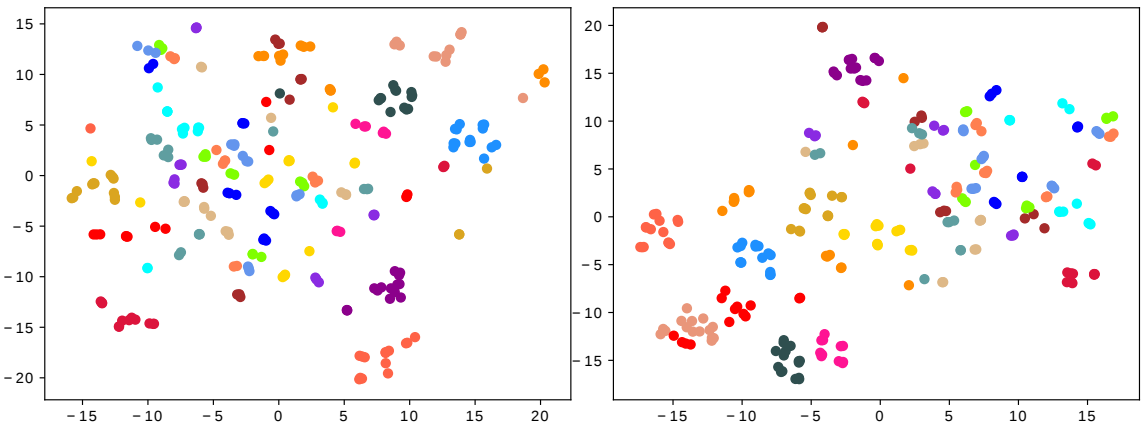
Figure A.4 – Network architecture for action recognition. The action prediction blocks can be repeated K times. The same architecture is used for pose and appearance recognition, except that for pose, each convolution uses half the number of features showed here. T corresponds the number of frames and N_a is the number of actions.

Appendix B

Additional Results and Experiments

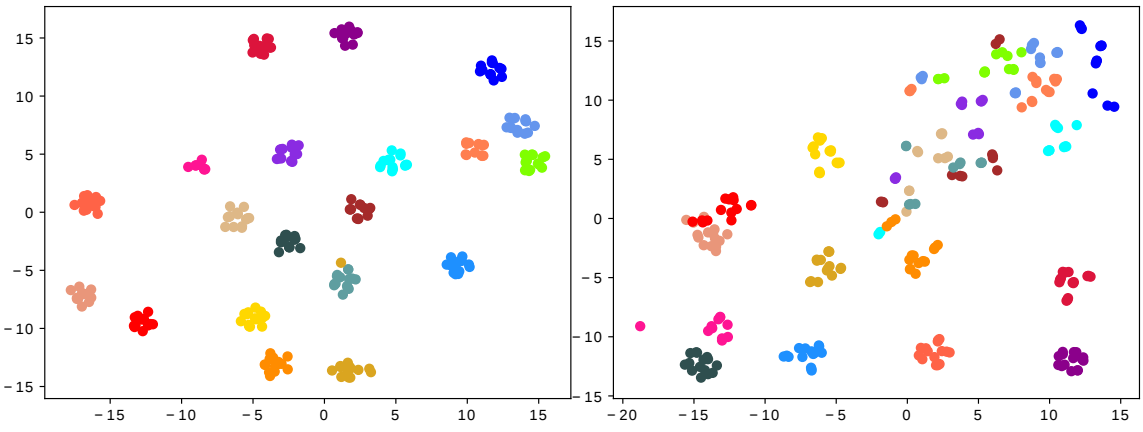
B.1 Feature Space for Skeleton Action Recognition

Considering the metric learning algorithm from section 3.2.3, in Figure B.1 we show a projection using t-SNE of the feature space before and after the linear transformation **L**, considering the MSR-Action3D dataset for action recognition.



(a) Global features (training samples)

(b) Global features (testing samples)



(c) Learned features (training samples)

(d) Learned features (testing samples)

Figure B.1 – t-SNE features projection for skeleton action recognition for MSR-Action3D.

Table B.1 – Results on LSP test samples using the PCK measure at 0.2 with OC annotations.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Avg. PCK
Detection based methods								
Kiefel and Gehler [66]	83.5	73.7	55.9	36.2	73.7	70.5	66.9	65.8
Ramakrishna et al. [113]	84.9	77.8	61.4	47.2	73.6	69.1	68.8	69.0
Pishchulin et al. [104]	87.5	77.6	61.4	47.6	79.0	75.2	68.4	71.0
Ouyang et al. [97]	86.5	78.2	61.7	49.3	76.9	70.0	67.6	70.0
Chen and Yuille [21]	91.5	84.7	70.3	63.2	82.7	78.1	72.0	77.5
Yang et al. [157]	90.6	89.1	80.3	73.5	85.5	82.8	68.8	81.5
Chu et al. [28]	93.7	87.2	78.2	73.8	88.2	83.0	80.9	83.6
Pishchulin et al. [105]	**97.4**	92.0	83.8	79.0	93.1	88.3	83.7	88.2
Regression based method								
Our method	**97.4**	**93.8**	**86.8**	**82.3**	**93.7**	**90.9**	**88.3**	**90.5**

Table B.2 – Results on LSP test samples using the PCP measure with OC annotations.

Method	Torso	Upper leg	Lower leg	Upper arm	Fore-arm	Head	PCP
Detection based methods							
Kiefel and Gehler [66]	84.3	74.5	67.6	54.1	28.3	78.3	61.2
Pishchulin et al. [103]	87.4	75.7	68.0	54.4	33.7	77.4	62.8
Ramakrishna et al. [113]	88.1	79.0	73.6	62.8	39.5	80.4	67.8
Ouyang et al. [97]	88.6	77.8	71.9	61.9	45.4	84.3	68.7
Pishchulin et al. [104]	88.7	78.9	73.2	61.8	45.0	85.1	69.2
Chen and Yuille [21]	92.7	82.9	77.0	69.2	55.4	87.8	75.0
Yang et al. [157]	96.5	88.7	81.7	78.8	66.7	83.1	81.1
Chu et al. [28]	95.4	87.6	83.2	76.9	65.2	89.6	81.1
Pishchulin et al. [105]	96.0	91.0	83.5	82.8	71.8	**96.2**	85.0
Regression based method							
Our method	**98.2**	**93.8**	**89.8**	**85.8**	**75.5**	96.0	**88.4**

B.2 Additional Results on Human Pose Estimation

Additional results for section 4.3.2, considering previous methods and older results, are given as follows. Results on LSP using PCK/OC metric are shown in Table B.1. Results on LSP using PCP/OC metric are shown in Table B.2. Results on the same metric considering PC annotations are provided in Table B.3 and Table B.4. Results on MPII using the PCKh metric are shown in Table B.5.

In Table B.6, we present a full comparison with previous results from related methods on Human3.6M, considering the validation set.

Table B.3 – Results on LSP test samples using the PCK measure at 0.2 with PC annotations.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	PCK
Detection based methods								
Pishchulin et al. [104]	87.2	56.7	46.7	38.0	61.0	57.5	52.7	57.1
Chen and Yuille [21]	91.8	78.2	71.8	65.5	73.3	70.2	63.4	73.4
Fan et al. [37]	92.4	75.2	65.3	64.0	75.7	68.3	70.4	73.0
Tompson et al. [138]	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.3
Yang et al. [157]	90.6	78.1	73.8	68.8	74.8	69.9	58.9	73.6
Rafi et al. [111]	95.8	86.2	79.3	75.0	86.6	83.8	79.8	83.8
Yu et al. [164]	87.2	88.2	82.4	76.3	91.4	85.8	78.7	84.3
Belag. and Ziss. [11]	95.2	89.0	81.5	77.0	83.7	87.0	82.8	85.2
Lifshitz et al. [77]	96.8	89.0	82.7	79.1	90.9	86.0	82.5	86.7
Pishchulin et al. [105]	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Insafutdinov et al. [54]	97.4	92.7	87.5	84.4	91.5	89.9	87.2	90.1
Wei et al. [149]	97.8	92.5	87.0	83.9	91.5	90.8	89.9	90.5
Bulat and Tzimi. [13]	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Chu et al. [29]	**98.1**	**93.7**	**89.3**	**86.9**	**93.4**	**94.0**	**92.5**	**92.6**
Regression based methods								
Carreira et al. [16]	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Our method	**97.5**	**93.3**	**87.6**	**84.6**	**92.8**	**92.0**	**90.0**	**91.1**

Table B.4 – Results on LSP test samples using the PCP measure with PC annotations.

Method	Torso	Upper leg	Lower leg	Upper arm	Fore-arm	Head	PCP
Detection based methods							
Pishchulin et al. [104]	88.7	63.6	58.4	46.0	35.2	85.1	58.0
Tompson et al. [138]	90.3	70.4	61.1	63.0	51.2	83.7	66.6
Fan et al. [37]	95.4	77.7	69.8	62.8	49.1	86.6	70.1
Chen and Yuille [21]	96.0	77.2	72.2	69.7	58.1	85.6	73.6
Yang et al. [157]	95.6	78.5	71.8	72.2	61.8	83.9	74.8
Rafi et al. [111]	97.6	87.3	80.2	76.8	66.2	93.3	81.2
Belag. and Ziss. [11]	96.0	86.7	82.2	79.4	69.4	89.4	82.1
Yu et al. [164]	98.0	93.1	88.1	82.9	72.6	83.0	85.4
Lifshitz et al. [77]	97.3	88.8	84.4	80.6	71.4	94.8	84.3
Pishchulin et al. [105]	97.0	88.8	82.0	82.4	71.8	95.8	84.3
Insafutdinov et al. [54]	97.0	90.6	86.9	86.1	79.5	95.4	87.8
Wei et al. [149]	98.0	92.2	89.1	85.8	77.9	95.0	88.3
Bulat and Tzimi. [13]	97.7	92.4	89.3	86.7	79.7	95.2	88.9
Chu et al. [29]	**98.4**	**95.0**	**92.8**	**88.5**	**81.2**	95.7	**90.9**
Regression based methods							
Carreira et al. [16]	95.3	81.8	73.3	66.7	51.0	84.4	72.5
Our method	**98.2**	**93.6**	**91.0**	**86.6**	**78.2**	**96.8**	**89.4**

Table B.5 – Comparison results with state-of-the-art methods on the MPII dataset on testing, using PCKh measure with threshold as 0.5 of the head segment length. Detection based methods are shown on top and regression based methods on bottom.

Method	Head	Shouler	Elbow	Wrist	Hip	Knee	Ankle	Total
Detection based methods								
Pishchulin et al. [104]	74.3	49.0	40.8	34.1	36.5	34.4	35.2	44.1
Tompson et al. [138]	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6
Tompson et al. [137]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Hu and Ramanan [51]	95.0	91.6	83.0	76.6	81.9	74.5	69.5	82.4
Pishchulin et al. [105]	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Lifshitz et al. [77]	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Gkioxary et al. [42]	96.2	93.1	86.7	82.1	85.2	81.4	74.1	86.1
Rafi et al. [111]	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3
Belagiannis and Ziss. [11]	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1
Insafutdinov et al. [54]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Wei et al. [149]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat and Tzimiropoulos [13]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Newell et al. [93]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Chu et al. [29]	**98.5**	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chou et al. [27]	98.2	**96.8**	92.2	88.0	**91.3**	89.1	84.9	91.8
Chen et al. [22]	98.1	96.5	**92.5**	**88.5**	90.2	**89.6**	**86.0**	**91.9**
Regression based methods								
Rogez et al. [117]	–	–	–	–	–	–	–	74.2
Carreira et al. [16]	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Sun et al. [130]	97.5	94.3	87.0	81.2	86.5	78.5	75.4	86.4
Our method	**98.1**	**96.6**	**92.0**	**87.5**	**90.6**	**88.0**	**82.7**	**91.2**

Table B.6 – 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
Chen and Ramanan [19]	89.8	97.5	89.9	107.8	107.3	93.5	136.0	133.1
Tekin et al. [135]	85.0	108.8	84.4	8.9	119.4	98.5	93.8	73.8
Tome et al. [136]	65.0	73.5	76.8	86.4	86.3	68.9	74.8	110.2
Zhou et al. [173]	68.7	74.8	67.8	76.4	76.3	84.0	70.2	88.0
Pavlakos et al. [99]	67.4	71.9	66.7	69.1	71.9	65.0	68.3	83.7
Mehta et al. [88]⋆	52.5	63.8	55.4	62.3	71.8	52.6	72.2	86.2
Martinez et al. [87]	51.8	56.2	58.1	59.0	69.5	55.2	58.1	74.0
Sun et al. [131]	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7
Ours (single-crop)	**51.5**	**53.4**	**49.0**	**52.5**	**53.9**	**50.3**	54.4	**63.6**
Ours (multi-crop + h.flip)	**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
Chen and Ramanan [19]	240.1	106.6	139.1	106.2	87.0	114.0	90.5	114.1
Tekin et al. [135]	170.4	85.1	95.7	116.9	62.1	113.7	94.8	100.1
Tome et al. [136]	173.9	85.0	110.7	85.8	71.4	86.3	73.1	88.4
Zhou et al. [173]	113.8	78.0	98.4	90.1	62.6	75.1	73.6	79.9
Pavlakos et al. [99]	96.5	71.4	76.9	65.8	59.1	74.9	63.2	71.9
Mehta et al. [88]⋆	120.0	66.0	79.8	63.9	48.9	76.8	53.7	68.6
Martinez et al. [87]	94.6	62.3	78.4	59.1	49.5	65.1	52.4	62.9
Sun et al. [131]	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
Ours (single-crop)	**73.5**	**55.3**	**61.9**	**50.1**	**46.0**	**60.2**	**51.0**	**55.1**
Ours (multi-crop + h.flip)	**70.9**	**53.7**	**60.3**	**48.9**	**44.4**	**57.9**	**48.9**	**53.2**

⋆ Method not using ground-truth bounding boxes.

Bibliography

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, Jan 2006. 9

[2] 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. 39, 42, 52, 75

[3] 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. 39, 45

[4] 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. 7

[5] AsusTek Computer Inc. Xtion PRO. 6

[6] A. Baak, M. Müller, G. Bharaj, H. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *2011 International Conference on Computer Vision*, pages 1092–1099, Nov 2011. 6

[7] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In A. A. Salah and B. Lepri, editors, *Human Behavior Understanding*, pages 29–39, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 13

[8] F. Baradel, C. Wolf, and J. Mille. Pose-conditioned spatio-temporal attention for human action recognition. *arxiv*, 1703.10106, 2017. xviii, 12, 67, 69, 70, 77, 81

[9] 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. 13, 81, 84

[10] 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. 7, 10

[11] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. *CoRR*, abs/1605.02914, 2016. 57, 97, 98

[12] L. Bottou. Stochastic Learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004. 22

[13] A. Bulat and G. Tzimiropoulos. Human pose estimation via Convolutional Part Heatmap Regression. In *European Conference on Computer Vision (ECCV)*, pages 717–732, 2016. 7, 10, 38, 42, 43, 97, 98

[14] C. Cao, Y. Zhang, C. Zhang, and H. Lu. Body joint guided 3d deep convolutional descriptors for action recognition. *CoRR*, abs/1704.07160, 2017. 13, 77, 81, 84

[15] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 10

[16] 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. 8, 42, 43, 97, 98

[17] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, July 2017. 13

[18] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 26

[19] 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. 9, 32, 47, 99

[20] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *3D Vision (3DV)*, 2016. 56

[21] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 96, 97

[22] 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. 32, 43, 48, 98

[23] 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. 8

[24] G. Ch'eron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In *ICCV*, 2015. 13, 67

[25] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 37, 50, 72

[26] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015. 36

[27] C. Chou, J. Chien, and H. Chen. Self adversarial training for human pose estimation. *CoRR*, abs/1707.02439, 2017. 8, 43, 48, 98

[28] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 42, 96

[29] 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. 7, 8, 10, 37, 38, 42, 43, 48, 97, 98

[30] R. Dabral, A. Mundhada, U. Kusupati, S. Afaque, A. Sharma, and A. Jain. Learning 3d human pose from structure and motion. In *The European Conference on Computer Vision (ECCV)*, September 2018. 9

[31] 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. 7

[32] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the VLAD image representation. In *ACM Multimedia*, Barcelona, Spain, Oct. 2013. 21

[33] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. 3D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold. *IEEE Transactions on Cybernetics*, Aug 2014. 12, 24

[34] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, Oct 2005. 13

[35] 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. 13, 81, 84

[36] T. Durand, N. Thome, and M. Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4743–4752, June 2016. 70

[37] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 32, 97

[38] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 9, 32

[39] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 39

[40] R. G. Freedman, H.-T. Jung, and S. Zilberstein. Temporal and object relations in unsupervised plan and activity recognition. In *AI for HRI: Papers from the AAAI 2015 Fall Symp*, pages 48–50, 2015. 17

[41] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 755–762, June 2010. 6

[42] G. Gkioxari, A. Toshev, and N. Jaitly. Chained Predictions Using Convolutional Neural Networks. *European Conference on Computer Vision (ECCV)*, 2016. 7, 10, 98

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 8

[44] A. Grinciunaite, A. Gudi, H. E. Tasli, and M. den Uyl. Human pose estimation in space and time using 3d CNN. *CoRR*, abs/1609.00036, 2016. 61

[45] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, Oct 2013. 6

[46] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 1

[47] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 32

[48] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[49] 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. 11

[50] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 50

[51] P. Hu and D. Ramanan. Bottom-up and top-down reasoning with convolutional latent-variable models. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. 98

[52] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: Articulated multi-person tracking in the wild. In *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 1293–1301, Honolulu, HI, USA, 2017. IEEE. 9, 10

[53] E. Insafutdinov and A. Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NeurIPS*, 2018. 1

[54] 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. 7, 10, 97, 98

[55] C. Ionescu, J. Carreira, and C. Sminchisescu. Iterated second-order label sensitive pooling for 3d human pose estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1661–1668, June 2014. 9

[56] C. Ionescu, F. Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 2220–2227, Nov 2011. 9

[57] 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. 9, 45, 51, 58, 61, 75

[58] U. Iqbal and J. Gall. Multi-person pose estimation with local joint-to-person associations. *CoRR*, abs/1608.08526, 2016. 10

[59] U. Iqbal, M. Garbade, and J. Gall. Pose for action - action for pose. *FG-2017*, 2017. 67, 77

[60] U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 10

[61] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010. 20

[62] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(9):1704–1716, Sep 2012. 21

[63] 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. 13

[64] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *International Conference on Computer Vision (ICCV)*, 2007. 13

[65] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. 39

[66] M. Kiefel and P. V. Gehler. *Human Pose Estimation with Fields of Parts*, pages 331–346. Springer International Publishing, Cham, 2014. 96

[67] 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. 67

[68] L. Ladicky, P. H. S. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 32

[69] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 13

[70] K. Lee, I. Lee, and S. Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *The European Conference on Computer Vision (ECCV)*, September 2018. 9

[71] E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, and B. Andres. Joint graph decomposition & node labeling: Problem, algorithms, applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 10

[72] M. Li and H. Leung. Graph-based approach for 3D human skeletal action recognition. *Pattern Recognition Letters*, pages –, 2016. 12, 24

[73] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II*, pages 332–347, 2014. 9

[74] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 9

[75] W. Li, Z. Zhang, and Z. Liu. Action Recognition Based on a Bag of 3D Points. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9–14, 2010. 11, 23

[76] B. Liang and L. Zheng. A Survey on Human Action Recognition Using Depth Sensors. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2015. 11

[77] I. Lifshitz, E. Fetaya, and S. Ullman. *Human Pose Estimation Using Deep Consensus Voting*, pages 246–260. Springer International Publishing, Cham, 2016. 7, 97, 98

[78] 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. 12

[79] 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. 12, 77

[80] 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. 81

[81] C. Lu, J. Jia, and C. K. Tang. Range-Sample Depth Feature for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 772–779, 2014. 11, 24

[82] J. Luo, W. Wang, and H. Qi. Group Sparsity and Geometry Constrained Dictionary Learning for Action Recognition from Depth Maps. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1809–1816, 2013. 12, 17, 19, 24

[83] 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. 9, 45, 63, 75, 76, 80, 81

[84] D. C. Luvizon, H. Tabia, and D. Picard. Human pose regression by combining indirect part detection and contextual information. *CoRR*, abs/1710.02322, 2017. 33, 34, 39, 74

[85] D. C. Luvizon, H. Tabia, and D. Picard. Learning features combination for human action recognition from skeleton sequences. *Pattern Recognition Letters*, 2017. 12

[86] V. Magnanimo, M. Saveriano, S. Rossi, and D. Lee. A Bayesian approach for task recognition and future human activity prediction. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 726–731, 2014. 17

[87] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 9, 99

[88] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*, 2017. 9, 47, 51, 52, 54, 58, 59, 63, 99

[89] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt. Single-shot multi-person 3d body pose estimation from monocular RGB input. *International Conference on 3D Vision (3DV)*, 2018. 10, 11, 89

[90] 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. volume 36, 2017. 9, 32, 54

[91] Microsoft Corp. Redmond WA. Kinect for Xbox 360. 6

[92] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2277–2287. Curran Associates, Inc., 2017. 10

[93] A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. *European Conference on Computer Vision (ECCV)*, pages 483–499, 2016. 7, 8, 9, 10, 33, 37, 38, 42, 43, 48, 50, 67, 98

[94] A. Nibali, Z. He, S. Morgan, and L. Prendergast. 3d human pose estimation with 2d marginal heatmaps, 2018. 9

[95] 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. 7, 48

[96] O. Oreifej and Z. Liu. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723, June 2013. 11, 24

[97] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2344, June 2014. 96

[98] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. 2017. 10

[99] 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. 9, 32, 47, 50, 54, 67, 84, 99

[100] A. Perez, H. Tabia, D. Declercq, and A. Zanotti. Using the conflict in dempster-shafer evidence theory as a rejection criterion in classifier output combination for 3d human action

recognition. *Image and Vision Computing*, 55:149 – 157, 2016. Handcrafted vs. Learned Representations for Human Action Recognition. 11

[101] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *International Conference on Computer Vision (ICCV)*, 2015. 7

[102] 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. 7, 8

[103] 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. 7, 96

[104] L. Pishchulin, M. Andriluka, P. V. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 3487–3494, 2013. 32, 96, 97, 98

[105] 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. 7, 10, 42, 96, 97, 98

[106] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3178–3185, June 2012. 10

[107] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn. Posebits for monocular human pose estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2352, 2014. 9

[108] 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. 9, 61

[109] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010. 11

[110] L. L. Presti and M. L. Cascia. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130–147, 2016. 11

[111] U. Rafi, I. Kostrikov, J. Gall, and B. Leibe. An efficient convolutional network for human pose estimation. In *BMVC*, volume 1, page 2, 2016. 7, 97, 98

[112] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian. Histogram of Oriented Principal Components for Cross-View Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, PP(99):1–1, 2016. 11, 24

[113] V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh. Pose Machines: Articulated Pose Estimation via Inference Machines. In *European Conference on Computer Vision (ECCV)*, 2014. 96

[114] M. Rayat Imtiaz Hossain and J. J. Little. Exploiting temporal information for 3d human pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 9

[115] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 11

[116] G. Rogez and C. Schmid. Image-based Synthesis for Deep 3D Human Pose Estimation. *International Journal of Computer Vision*, 126(9):993–1008, Sept. 2018. 10

[117] 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. 10, 43, 48, 63, 89, 98

[118] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *CoRR*, abs/1803.00455, 2018. 56

[119] B. Safadi, N. Derbas, and G. Quénot. Descriptor Optimization for Multimedia Indexing and Retrieval. *Multimedia Tools and Applications (MTAP)*, 74(4):1267–1290, Feb 2015. 21

[120] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3674–3681, Washington, DC, USA, 2013. IEEE Computer Society. 39

[121] 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*, 152:1 – 20, 2016. 6

[122] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala. Recognizing Actions from Depth Cameras as Weakly Aligned Multi-part Bag-of-Poses. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 479–485, 2013. 23, 24

[123] 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. 76

[124] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+d videos. *TPAMI*, 2017. 12, 77, 84

[125] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time Human Pose Recognition in Parts from Single Depth Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '11, pages 1297–1304, 2011. 6, 17

[126] M. Siddiqui and G. Medioni. Human pose estimation from a single view point, real-time range sensor. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 1–8, June 2010. 6

[127] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 32

[128] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava. Accurate 3D action recognition using learning on the Grassmann manifold. *Pattern Recognition*, 48(2):556 – 567, 2015. 12, 24, 27

[129] 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. 12

[130] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. *arXiv preprint arXiv:1702.07432*, 2017. 8, 43, 98

[131] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. *arXiv preprint arXiv:1702.07432*, 2017. 9, 11, 47, 48, 57, 63, 80, 99

[132] 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. 9, 33, 63, 80

[133] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. 37, 38, 68

[134] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*, 2016. 9

[135] 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. 9, 99

[136] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *CVPR*, July 2017. 9, 99

[137] 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. 7, 41, 98

[138] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1799–1807. Curran Associates, Inc., 2014. 97, 98

[139] A. Toshev and C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014. 7, 8, 32

[140] Q. D. Tran and N. Q. Ly. An effective fusion scheme of spatio-temporal features for human action recognition in RGB-D video. In *International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 246–251, 2013. 11, 17, 24

[141] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018. 13

[142] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 10

[143] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential Recurrent Neural Networks for Action Recognition. In *IEEE International Conference on Computer Vision (ICCV)*, December 2015. 12, 24

[144] R. Vemulapalli, F. Arrate, and R. Chellappa. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, 2014. 12, 17, 24

[145] 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. 13

[146] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, May 2013. 13

[147] H. Wang and C. Schmid. Action recognition with improved trajectories. In *2013 IEEE International Conference on Computer Vision*, pages 3551–3558, Dec 2013. 13

[148] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, June 2012. 11, 17, 18, 19, 23, 24

[149] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7, 10, 97, 98

[150] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, 31(6):188:1–188:12, Nov. 2012. 6

[151] K. Weinberger and L. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *The Journal of Machine Learning Research (JMLR)*, 10:207–244, 2009. 21

[152] D. Weinland, R. Ronfard, and E. Boyer. A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition. *Computer Vision and Image Understanding*, 115(2):224–241, feb 2011. 17

[153] L. Xia, C.-C. Chen, and J. K. Aggarwal. View Invariant Human Action Recognition Using Histograms of 3D Joints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20–27. IEEE, 2012. 11, 17, 23, 24

[154] 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. 13, 76

[155] 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. 8, 48

[156] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang. Learning feature pyramids for human pose estimation. In *arXiv preprint arXiv:1708.01101*, 2017. 32

[157] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 96, 97

[158] W. Yang, W. Ouyang, X. Wang, 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)*, June 2018. 9, 80, 84

[159] X. Yang and Y. Tian. Super Normal Vector for Activity Recognition Using Depth Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 804–811, 2014. 11, 24

[160] Y. Yang, S. Baker, A. Kannan, and D. Ramanan. Recognizing proxemics in personal photos. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3522–3529, June 2012. 32

[161] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, Dec 2013. 39

[162] 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. 67

[163] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. *European Conference on Computer Vision (ECCV)*, 2016. 34

[164] X. Yu, F. Zhou, and M. Chandraker. *Deep Deformation Network for Object Landmark Localization*, pages 52–70. Springer International Publishing, Cham, 2016. 97

[165] A. Zanfir, E. Marinoiu, and C. Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes - the importance of multiple scene constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 10, 11, 61, 89

[166] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2752–2759, 2013. 12

[167] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang. RGB-D-based action recognition datasets: A survey. *Pattern Recognition*, 60:86 – 105, 2016. 23

[168] 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. 76

[169] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, Feb 2012. 6

[170] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 10, 54, 63

[171] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. *Computer Vision ECCV 2016 Workshops*, 2016. 9

[172] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 9

[173] 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. 9, 99

[174] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*, February 2016. 12

[175] Y. Zhu, W. Chen, and G. Guo. Fusing Spatiotemporal Features and Joints for 3D Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 486–491, 2013. 24

[176] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005. 41