Towards end-to-end training of proposal-based 3D human pose estimation

Daniel Ajisafe[1], **Joseph Domguia**[1], **James Tang**[2], **Lynn Ellenberger**[3,4], **Jörg Spörri**[3,4], **Helge Rhodin**[2]

[1]African Masters in Machine Intelligence (AMMI),
African Institute for Mathematical Sciences, Rwanda
[2]Department of Computer Science, University of British Columbia, Canada
[3]Sports Medical Research Group, Department of Orthopaedics, Balgrist University Hospital, University of Zurich, Zurich, Switzerland
[4]University Centre for Prevention and Sports Medicine, Department of Orthopaedics, Balgrist University Hospital, University of Zurich, Zurich, Switzerland

Abstract

2D Pose estimation is very important in localizing human joints such as elbows, hip and foot, in images or videos. While 2D pose estimation estimates the location of keypoints in 2D space relative to an image or video frame, 3D pose estimation from an input image is difficult because the depth information that is lost as objects and points in 3D space are mapped onto a 2D image plane, is hard to recover. Hence, computing properties of the 3D world from visual data is hard. In this work, we demonstrate single-person 3D pose estimation from indoor videos on a custom dataset, with the long-term goal to enable motion analysis for injury prevention in sports. We present three different frameworks for directly estimating 3D pose from observed video frames up to stage-wise training, two of which are trained using end-to-end learning. Empirically, we observe average precision of 99% on 2D human pose and up to 90% for 3D pose estimation.

1 Introduction

Computer vision tasks like image classification, segmentation and object detection have largely been addressed from a 2D perspective. The world, however, is 3D and it will be essential for models to have a 3D understanding of their environments in order to successfully interpret, and navigate within the real world. 3D human pose estimation remains one of the significant components that tries to solve this problem but remains a core challenge in computer vision. This is due to a number of factors to which the model needs to be invariant to, such as the background scene, clothing texture and shape, lightening condition and more importantly, the depth information that is lost when we project 3D properties onto a 2D plane.

A lot of work has been done in estimating 3D human pose in images and videos. These includes high-performing convolutional based methods such as [1, 2]. Before deep convolutional neural networks came into the scene, traditional methods like [3, 4] and [5] have detected 2D pose using pictorial structures and trained part-based models which can inform 3D pose estimation.

With the deep learning regime, popular architectures like OpenPose [6] followed by Maskrcnn [7] have made siginificant breakthroughs in pose estimation and remains the most commonly used meth-

Workshop on Differentiable Vision, Graphics, and Physics in Machine Learning at NeurIPS 2020.

ods in 2D estimation. OpenPose is a bottom-up approach that detect joints without needing a person detector while Maskrcnn is an efficient top-down approach that uses a person detector to output 2D joints.

In this paper, we follow this top-down approach using Maskrcnn [7]. Though this approach is a high-performing method, it still suffers from a number of issues such as non-differentiability in combination with other systems and an extra post-processing overhead for pose estimation. Both issues makes it difficult to train end-to-end systems. We solve this problem by defining a soft-argmax function which is similar to [8] in order to train end-to-end systems and extend the Maskrcnn architecture to 3D pose estimation.

Our main contribution, therefore, is extending the work of He et al. [7] in Maskrcnn with integral pose regression introduced by [8] for 2D and 3D pose estimation on a new dataset. Specifically, we performed a comparative analysis on three different variants, and achieved a surpassing result higher than our baseline models on the dataset. Hereinafter, these variants are also referred to as frameworks. Our test set is a dataset of jumping motions that, with the right reconstruction algorithm, is set up to enable injury prevention in sports.

2 Related Work

3D human pose estimation has been an heavily researched area in the field of computer vision for quite a long time. In this section, we would describe some past and recent work in this area and expound on their relation to our study.

2-step vs 1-step methods: Different approaches for estimating 3D pose can be broadly classified into two major categories, which is i) predicting relative 3D pose by first detecting 2D keypoints or ii) directly estimating 3D pose without an intermediate supervision.

The former approach (2-step) somewhat involves a top-down approach because it first proposes regions of interest (ROI) using person detectors in the first step before detecting 2D keypoints in each ROI. Many methods [7, 9–12] take advantage of powerful person detectors to estimate these 2D keypoints. For example, Cascaded Pyramid Network (CPN) adopts this top-down pipeline by first generating a set of human bounding boxes based on a detector, followed by a Global Pyramid Network that successfully recognize occluded or invincible keypoints. Maskrcnn follows the same method in predicting 2D keypoints. It extends a proposal-based backbone by first proposing feature maps, and then estimates keypoints using a parallel keypoint prediction branch.

At the end, these 2D keypoints with good detectors are lifted to estimate 3D pose. Martinez et al [13] used a relatively simple deep feed forward network to lift both detected and groundtruth 2D joint locations to 3D space. Moreno-Noguer [14] addressed the problem of 3D human pose estimation as a 2D-to-3D distance matrix regression. Chen et al [15] reasoned through intermediate 2D pose predictions to estimate 3D pose, by performing a simple nearest neighbor search given a 3D pose library and a large number of generated 2D projections, whereas Zhou et al [2] simply predicts the 3D pose by regressing the depth from a set of given 2D keypoints.

On the other hand, the latter approach (1-step) estimates 3D pose directly from image features [16–20]. Agarwal et al [16] predicted 3D pose using direct nonlinear regression given a set of shape descriptor vectors extracted automatically from image silhouettes, whereas, Rogez et al [17] tackles 3D pose estimation from direct image features by recursively clustering and merging classes on a decision tree. However, the best performing methods in most literature, use the 2-step approach to learn from 2D keypoint trajectories.

Regression vs Heatmap-based methods: Several approaches have addressed pose estimation as either a keypoint regression [21–24] or heatmap detection problem [25–30]. In the former, Toshev et al. [21] used deep neural networks (DNN) to provide an holistic reasoning for pose regression. The author use a cascade of DNN regressors to estimate joint locations for even joints that are barely visible. Alternatively, Sun et al. [22] formulate the problem from a structural perspective. It estimates 3D pose estimation using a structure-aware regression approach, which according to the study uses the bone information instead of joints to estimate 3D pose. To the best of our knowledge, all methods using regression-only approach without reasoning about the spatial structure have been surpassed by heatmap based methods.

In the latter, heatmap based methods have shown tremendous progress in pose detection [25–30]. For example, Chu et al. [25] use stacked hourglass networks to generate attention maps from features at multiple resolutions. The attention maps where used to predict heatmaps for different body locations. Bulat et al. [27] use a cascade of convolutional neural networks (CNN) to output part-detection heatmaps. However, heatmap based method requires i) a post-processing step at inference time ii) creating artificial groundtruth heatmaps from joint locations using for example, gaussian distributions, and iii) inability to combine other architectures with it due to its non-differentiability. A number of methods [8, 31, 32] solve this problem by introducing an integral regression. Our approach has similarities to these methods, however, unlike their models, our work explore multiple frameworks for 3D pose estimation by extending the work of He et al. [7].

3 Experimental Setup

3.1 Dataset

The dataset was manually collected for an initial study describing the dynamic knee valgus of professional alpine (sport) skiers during Drop Jump landings (DJ) and single-leg squats (SLS) [33]. Elite competitive skiers and Youth competitive alpine skiers were examined for their maximal knee displacement (MKD) using a marker-based 3D motion analysis in evaluating the dynamic knee valgus.

In our own study, we estimate the relative 2D and 3D human pose that matches the spatial information of the detected athlete in each video frame. Like the medial knee displacement [33] is estimated using some keypoints, we also use the same keypoints to estimate 2D and 3D information. In this study, these keypoints are considered to be the relevant joints.

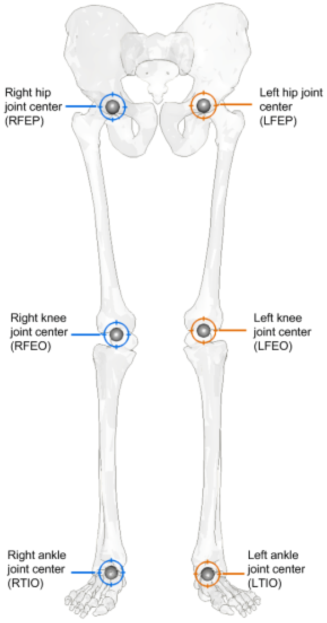

Figure 1: Relevant Plug-in Gait keypoints or joint centers

Name	Visual Description	Plug-in Gait Description
RFEP	Right hip joint center	Right Femur Proximal
LFEP	Left hip joint center	Left Femur Proximal
RFEO	Right knee joint center	Right Femur Origin
LFEO	Left knee joint center	Left Femur Origin
RTIO	Right ankle joint center	Right Tibia Origin
LTIO	Left ankle joint center	Left Tibia Origin

The dataset contains 3 sets of recordings, which are Set-A (with 42 elite skiers), Set-B1 (with 118 youth skiers in year 1), and Set-B2 (with 161 youth skiers in year 2). About 102 youth skiers from year 1 (Set-B1) also appear in year 2 (Set-B2). Each athlete in the dataset has a recording of 1-2 videos. Some recordings that were empty or whose calibration file were without relevant keypoints were dropped, making it a total of 587 recordings.

The dataset was splitted proportionally in a 70:15:15 ratio and we ensured that the athletes who participated both in Set-B1 and Set-B2 are in the same sets. This is to avoid overlapping during training, validation and testing. The average no of frames for each video is about 700 frames with an average duration of 6s. In training, validation and testing, we sample 100 frames with a stride of 3 for each video. All frameworks were trained using mini-batch of 3 samples, a learning rate of 0.001 and 15,000 iteration steps.

3.2 Pre-model Processing and Lens Distortion

Physical cameras use lenses, and images or videos taken with these cameras do not perfectly follow the pinhole camera model. Therefore, we account for lens distortion in the following sections by distorting groundtruth 2D keypoints with radial distortion parameters obtained by checkerboard calibration.

3.2.1 Radial distortion model

XCP file format is used to store camera calibration information for vicon motion capture systems. In our case, it contains the information required to perform lens distortion correction and to compute the projection matrix P. To be clear, the lens distortion correction takes raw 2D points and transforms them to 2D points which could be considered to have arisen via a pinhole camera. Most XCP files will have for each "camera" only one "keyframe" sub node containing the actual calibration information. In our system, we use the following attributes of a "camera" node:

- PIXEL_ASPECT_RATIO: the ratio of the physical dimensions of a pixel.
- SENSOR_SIZE: defined in pixels.
- SKEW

The important attributes of the "keyframe" node:

- ORIENTATION: 3D rotation using quaternion representation.
- POSITION: 3D translation.
- VICON_RADIAL or VICON_RADIAL2: containing the radial distortion information.
- IMAGE_ERROR: mean error in pixels which is the result of the calibration procedure.
- PRINCIPAL_POINT
- FOCAL_LENGTH

We use the following definition:

PIXEL_ASPECT_RATIO= "a"
PRINCIPAL_POINT = "x_{pp} y_{pp}"
VICON_RADIAL = "w_0 w_1" , meaning $[x_{dc}\ y_{dc}]$ = $[x_{pp}\ y_{pp}]$, and $w_2 = 0$ or:
VICON_RADIAL2 = Vicon3Parameter "x_{dc} y_{dc} w_0 w_1 w_2"

Note $[x_{dc}\ y_{dc}]$ is the distortion centre, the image point about which radial distortion occurs. This is generally assumed to be same as the principal point. In Vicon_radial2 we select only w_0 and w_1 in a similar correspondence to Vicon_radial.

The raw 2D point and corrected 2D point are, respectively, defined as

$$P_r = \begin{bmatrix} x_r \\ y_r \end{bmatrix} \text{ and } P_c = \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (1)$$

The correct radial distortion is calculated as

$$P_c = \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} s \cdot d_x + x_{pp} \\ (s \cdot d_y + y_{pp})/a \end{bmatrix}, \tag{2}$$

where the radius is calculated as $r = \|dp\|$, the scale factor is calculated as $s = 1 + w_0 r^2 + w_1 r^4 + w_2 r^6$, and

$$d_p = \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} x_r - x_{dc} \\ a(y_r - y_{dc}) \end{bmatrix}. \tag{3}$$

3.2.2 Projection matrix

We use the following definition:

POSITION = ”t” (3-vector)
ORIENTATION = ”q” (4-vector)
FOCAL_LENGTH = ”f”
SKEW = ”k”

First we convert the quaternion q to a 3x3 rotation matrix R after which we compose the 3x4 matrix P_a

$$Pa = [R - R_t]\,. \tag{4}$$

Next, we compose the 3x3 intrinsic matrix K:

$$K = \begin{bmatrix} f & k & x_{pp} \\ 0 & f/a & y_{pp} \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

Finally, the projected (3x4) matrix P is calculated as:

$$Pa = KP_a \tag{6}$$

We used this projection matrix to transform the 3D points into 2D points before distorting the 2D points onto the distorted image. To distort the 2D points, we calculate our radial distortion parameters by first centering the 2D points relative to the principal point/distortion centre.

$$d_p = (p_{px} - x_{pp}) \tag{7}$$

Then we calculate the radial distortion as a root-mean squared over the 2D coordinates before distortion. See Figure 3 with distorted keypoints.

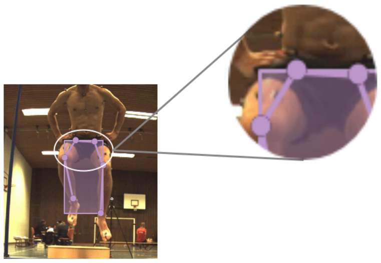

Figure 2: Sample Image with no distortion

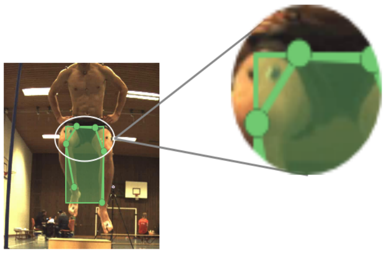

Figure 3: Sample Image with distortion

3.3 Normalization

We applied standard normalization to the groundtruth 3D points by subtracting the mean and dividing by the standard deviation. Seeing that, we do not estimate the world coordinates of the 3D pose, we center the 3D pose around the mid hip-joint which is the average of the right hip joint center (RFEP) and the left hip joint center (LFEP).

4 Methods and Implementation details

Inspired by the work of He et al. [7], our method extends Maskrcnn [7] in each of the different frameworks proposed. Therefore, we present only a brief overview of the Maskrcnn architecture and elaborate more on our extensions to the architecture.

Maskrcnn: is notable for performing extremely well in instance detection tasks. It is a general framework for object instance segmentation. Most of its architectural design follows the same as Faster R-CNN [34], except with a new pixel-pixel alignment called ROI Align and an additional mask branch for keypoint estimation. Our extensions are applied to this mask branch because it extracts rich spatial information for each detected object.

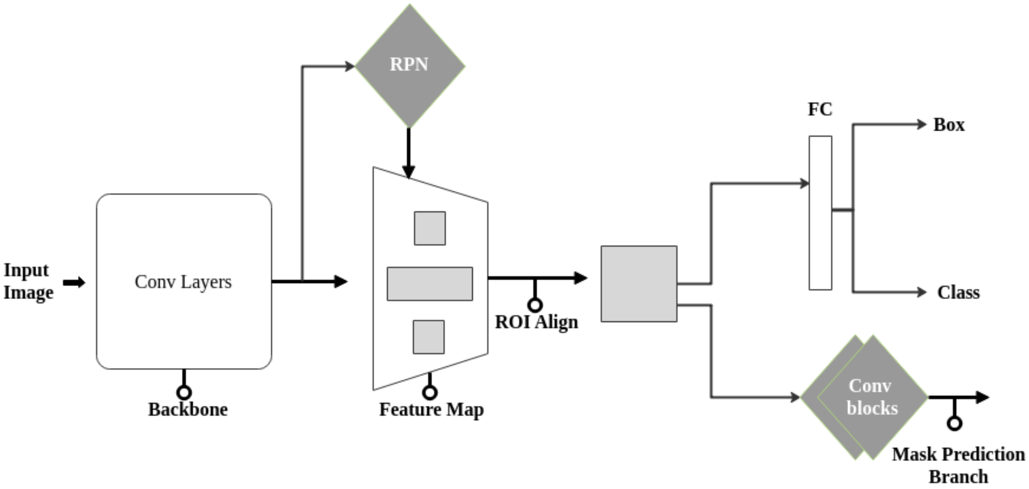

Figure 4: Network architecture for Maskrcnn

4.1 Integral 2D heatmap + Integral 3D Location map (End-to-End training)

In this framework, we adopt the same design as Maskrcnn [7] and extend it by first, estimating directly, the global 2D coordinates of an object from the spatial information coming from the mask branch in a differentiable way and also, simultaneously generate the 3D location of the body pose with additional location maps in a single step.

$$f : \mathbb{R}^{N \times H \times W \times 3} \rightarrow \mathbb{R}^{N \times K \times 2} \;\&\; \mathbb{R}^{N \times K \times 3} \tag{8}$$

To estimate joint 2D and 3D pose concurrently, we introduced additional heatmaps to Maskrcnn called location maps similar to [35]. The proposed location maps X_i, Y_j and Z_k, captures the 3D information of detected joints in 3D space while the 2D heatmaps captures information about the 2D joints. To accommodate these new location maps, we modify the direct output head from Maskrcnn to produce 24 heatmaps, 6 for the 2D output and 18 for the 3D output. Each heatmap is a 56 x 56 pixel spatial output. While most approaches use direct regression on heatmaps, the results of these approaches are not entirely satisfactory. To also alleviate this problem, we defined an integral layer [8] which takes the expectation (soft-argmax) over the locations rather than the maximum (hard-argmax) . The x_i, y_j and z_k values for the 3D coordinates are estimated by weighting the respective location maps with the corresponding 2D heatmap H_m and taking a sum. While the expectation for 2D is also taken over its corresponding heatmap H_m. This integral step is fully differentiable and allows us to train the framework end-to-end on groundtruth 2D and 3D coordinates without having to create artificial groundtruth heatmaps.

The discrete form of the integral step for 2D is defined as:

$$y_k^{\text{2D}} = \sum_{p_y=1}^{H} \sum_{p_x=1}^{W} p.\tilde{H}_k(p) \in R^{N \times K \times 2} \tag{9}$$

$$= \{x \in \mathbb{R} \mid 0 < x < 1\} \tag{10}$$

and for 3D is defined as:

$$y_k^{\text{3D}} = \sum_{p_y=1}^{H} \sum_{p_x=1}^{W} L_k(p).\tilde{H}_k(p) \in R^{N \times K \times 3} \tag{11}$$

where L is the location map and L predicts one 3D position per position p. H_k is the normalized heatmap with non-negative values that sum to 1 and Ω is the domain of the normalized heatmap.

$$\tilde{H}_k(p) = \frac{\exp^{H_k(p)}}{\int_{q \in \Omega} \exp^{H_k(q)}} \in R^{N \times K \times S \times S} \tag{12}$$

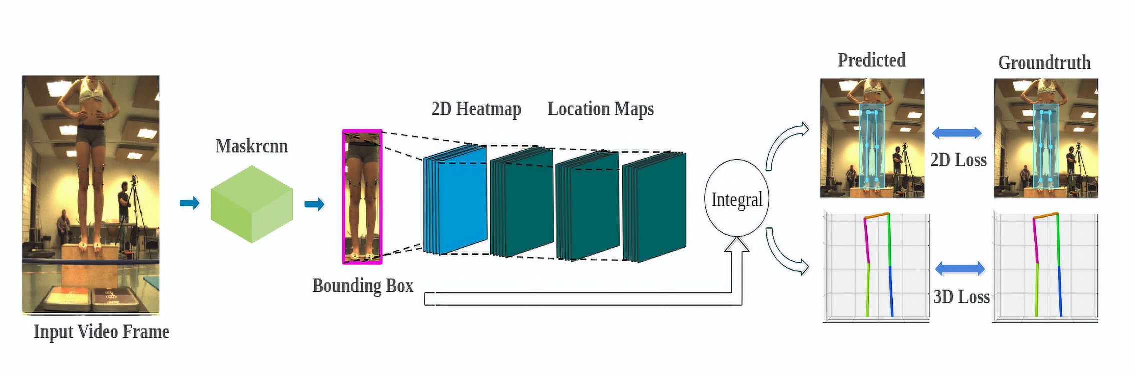

Figure 5: Integral 2D heatmap + Integral 3D Location map (End-to-End training)

We train this single architecture with a weighted loss $L = \alpha L_{2D} + \beta L_{3D}$ where L_{2D} and L_{3D} are MSE (mean squared error) loss functions, $\alpha = 0.7$, and $\beta = 0.3$. These two losses are normalized such that they fall relatively within the same scale.

$$L_{2D/3D} = \frac{1}{N_k} \sum_{i=1}^{N_k} (y_i - \hat{y}_i)^2 \tag{13}$$

where y_i and $\hat{y}_i$ are the groundtruth and predicted joints respectively.

4.2 Integral 2D heatmap + 3D MLP (Part training)

In this framework, we adopt the same design as Maskrcnn [7] and extend it with an integral step for 2D task only, and combined this modification with a 2nd multi-layer perceptron network [13] that is trained separately.

The first architecture which is a modified Maskrcnn is expected to take in the image evidence and output direct 2D coordinates following the integral step while the 2nd architecture, a multi-layer perceptron network lifts these 2D coordinates as input to estimate 3D pose. Specifically, the 2nd architecture uses a linear layer to map the input to a 1024 dimensional feature space and is followed by two residual blocks that includes a linear layer, batch norm, rectified linear units (ReLU) and a dropout. A final linear layer is used to regress the 3D pose from the feature space.

In this combined system, both architectures are trained separately. We first optimize the modified Maskrcnn architecture on 2D alone and store the optimal weights achieving top performance on 2D task. The 2D output is in the global coordinates relative to the input image. With this coordinates, the multi-layer perceptron network understands the global structure of the 2D points to estimate the 3D pose. Next, we initialize and freeze the same architecture with the pre-trained weights and perform 3D supervision on the multi-layer perceptron network. One of the inspiration for this approach is to verify if the initialization point of these weights would be a good starting point for the 3D task.

The Loss functions for 2D and 3D are also defined as:

$$L_{2D/3D} = \frac{1}{N_k} \sum_{i=1}^{N_k} (y_i - \hat{y}_i)^2 \tag{14}$$

where y_i and $\hat{y}_i$ are the groundtruth and predicted joints respectively.

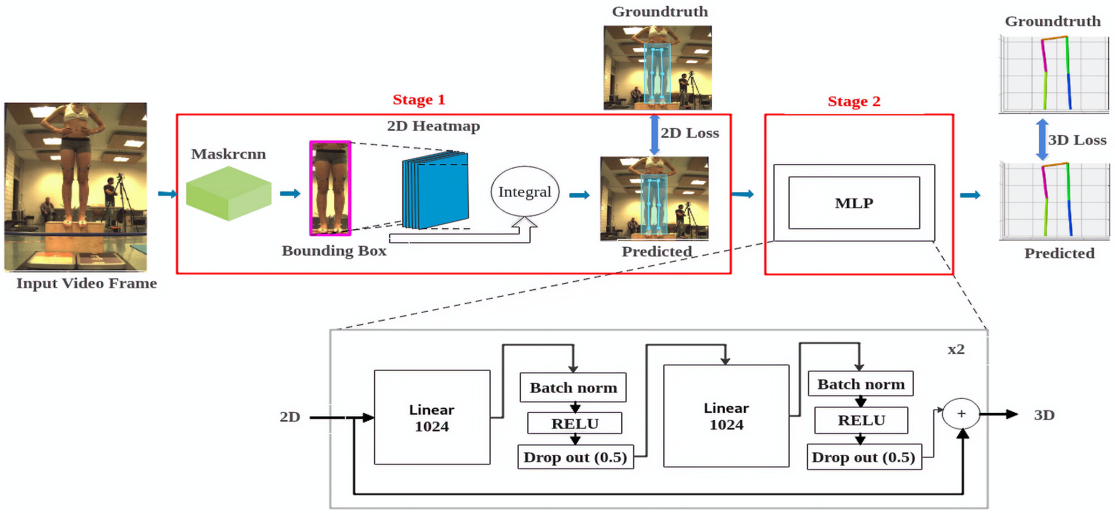

Figure 6: Integral 2D heatmap + 3D MLP (Part training)

4.3 Integral 2D heatmap + 3D MLP (End-to-End training)

In this framework, we combined the two architectures together, which is the same as Figure 6 but trained in an end-to-end fashion. Instead of initializing a part of the network with weights learned from 2D task, we jointly optimize 2D and 3D weights. In a sequential manner, the estimated 2D coordinates from the first architecture is passed as input to the multi-layer perceptron network for 3D pose estimation.

We trained this combined architecture jointly with a weighted keypoint loss $L = \alpha L_{2D} + \beta L_{3D}$ where L_{2D} and L_{3D} are MSE (mean squared error) loss functions, $\alpha = 0.7$, and $\beta = 0.7$.

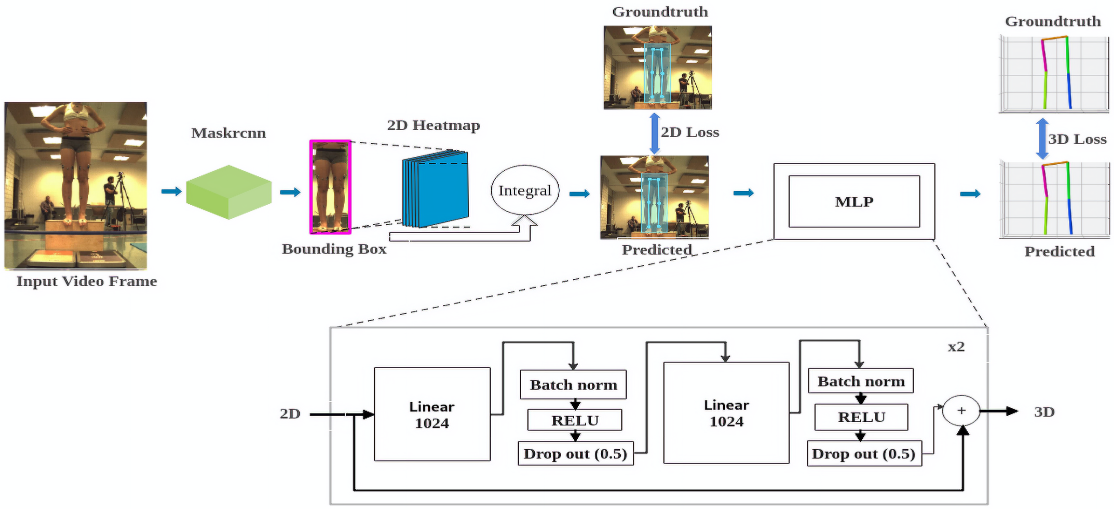

Figure 7: Integral 2D heatmap + 3D MLP (End-to-End training)

5 Results

5.1 Metrics

We use the Microsoft CoCo mean average precision (mAP) for evaluating the intermediate 2D pose and the MPI percentage of correctly positioned joints (PCK) for 3D evaluation. The PCK metric is the percentage of joints with predicted locations that are no further than the groundtruth by a certain threshold. Table 1 shows the results for different thresholds at 0.10m, 0.25m and 0.50m respectively.

5.2 Experimental results

In order to perform a comparative analysis on the three frameworks, we carry out a series of ablation studies with different configurations. Table 1 shows the results for PCK and Table 2 shows the results for 2D evaluation using mAP.

Framework	PCK@0.10m	PCK@0.25m	PCK@0.50m
Integral 2D heatmap + Integral 3D Location map (End-to-End training)	89.8	99.5	100.0
Integral 2D heatmap + 3D MLP (Part training)	88.6	99.5	99.8
Integral 2D heatmap + 3D MLP (End-to-End training)	79.9	99.4	100.0

Table 1: Results for 3D evaluation using Percentage of Correct Keypoint (PCK) accuracy

Framework	mAP
Integral 2D heatmap + Integral 3D Location map (End-to-End training)	99.0
Integral 2D heatmap + 3D MLP (Part training)	82.6
Integral 2D heatmap + 3D MLP (End-to-End training)	99.0

Table 2: Results for 2D evaluation using Mean Average Precision (mAP)

Our experiments show that our end-to-end Integral 3D Location map solution with (89.8% PCK) improves on the baselines (88.6% and 79.9% PCK) respectively at all thresholds. By directly estimating the 3D information we have better results, compared to multi-stage architectures where the depth information is somewhat lost in between.

6 Conclusion

In this work, we presented three different frameworks for directly estimating 3D pose from observed video frames on a custom dataset. In each of these frameworks, we explore differentiable 2D and 3D estimators and achieve the best performance on a single-step architecture. By directly estimating the 3D information we have better results, compared to multi-stage architectures where the depth information is somewhat lost in between.

The multi-stage architecture however, is posed to be more domain invariant as it can make use of both indoor and in the wild images to train its 2D estimators. These 2D estimators can become input for 3D pose estimation thereby making it invariant to a particular domain. However, estimating 3D pose from 2D joints is insufficient on a particular dataset because 2D pose data contains less information than images, thus inheriting more ambiguities.

We therefore conclude that the single-step Integral 3D location map approach benefits more from rich information that is contained within the image and performs very well on a particular dataset. By directly estimating the 3D information we have better results. Still, the attained accuracy is unfortunately below the error margins needed for the motion analysis application to facilitate injury prevention. At this stage, we recommend using multi-view approaches or solutions that incorporate domain knowledge.

For further research, it would be interesting to incorporate mixture density networks with integral pose regression, where we can generate more than a single hypothesis of the 3D pose on our dataset.

References

[1] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, “3D human pose estimation in video with temporal convolutions and semi-supervised training.” In CVPR, 2019. 1

[2] X. Zhou, Q. Huang, X. Sun, X. Xue, Y. Wei, “Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach.” In ICCV, 2017. 1, 2

[3] M. Dantone, J. Gall, C. Leistner, and L. Gool, “Human pose estimation using body parts dependent joint regressors.” In CVPR, 2013. 1

[4] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition.” In IJCV, 2005. 1

[5] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, “2d articulated human pose estimation and retrieval in (almost) unconstrained still images.” In IJCV, 2013. 1

[6] Z. Cao, G. Hidalgo, T. Simon, S. Wei, Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.” In CVPR, 2017. 1

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask rcnn.” In Computer Vision (ICCV), 2017 IEEE International Conference on, pages 2980-2988, 2017. 1, 2, 3, 6, 7

[8] X. Sun, B. Xiao, F. Wei, S. Liang , and Y. Wei, “Integral Human Pose Regression.” In ICCV, 2017. 2, 3, 6

[9] Y. Chen, Z. Wang, Y. Peng, Z. Zhan, G. Yu, and J. Sun, “Cascaded Pyramid Network for Multi-Person Pose Estimation.” In CVPR, 2018. 2

[10] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu., “Rmpe: Regional multi-person pose estimation.” In ICCV, 2017. 2

[11] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tomp- son, C. Bregler, and K. Murphy., “Towards Accurate Multi- person Pose Estimation in the Wild.” ArXiv e-prints, 2017. 2

[12] S. Huang, M. Gong, and D. Tao., “A coarse-fine network for keypoint localization.” In ICCV, 2017. 2

[13] J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A simple yet effective baseline for 3d human pose estimation.” In ICCV, 2017. 2, 7

[14] F. Moreno-Noguer, “3D Human Pose Estimation from a Single Image via Distance Matrix Regression.” In CVPR, 2017. 2

[15] C. Chen, D. Ramanan, “3D Human Pose Estimation = 2D Pose Estimation + Matching.” In CVPR, 2017. 2

[16] A. Agarwal and B. Triggs, “3D human pose from silhouettes by relevance vector regression.” In CVPR, 2004. 2

[17] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. S. Torr, “Randomized trees for human pose detection.” In CVPR, 2008. 2

[18] C. Sminchisescu, A. Kanaujia, Z. Li, and D. N. Metaxas, “Generative modeling for continuous non-linearly embedded visual inference.” In CVPR, 2005. 2

[19] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, “Fast algorithms for large scale conditional 3D prediction.” In CVPR, 2008. 2

[20] G. Shakhnarovich, P. Viola, and T. Darrell, “Fast pose estimation with parameter-sensitive hashing.” In CVPR, 2003. 2

[21] A. Toshev and C. Szegedy., “DeepPose: Human Pose Esti- mation via Deep Neural Networks.” In CVPR, 2014. 2

[22] X. Sun, J. Shang, S. Liang, and Y. Wei, “Compositional human pose regression.” In ICCV, 2017. 2

[23] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, “Deep convolutional neural networks for efficient pose estimation in gesture videos..” In ACCV, 2014. 2

[24] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback.” In CVPR, 2016. 2

[25] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X, “Multi-context attention for human pose estima- tion.” In CVPR, 2017. 2, 3

[26] A. Newell, K. Yang, and J. Deng, “Stacked Hourglass Net- works for Human Pose Estimation.” In ECCV, 2016. 2, 3

[27] A. Bulat and G. Tzimiropoulos, “Human pose estimation via Convolutional Part Heatmap Regression.” In ECCV, 2016. 2, 3

[28] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, “Robust optimization for deep regression.” In ICCV, 2015. 2, 3

[29] U. Rafi, I. Kostrikov, J. Gall, and B. Leibe, “An efficient convolutional network for human pose estimation.” In BMVC, 2016. 2, 3

[30] G. Gkioxari, A. Toshev, and N. Jaitly, “Chained Predictions Using Convolutional Neural Networks..” In ECCV, 2016. 2, 3

[31] D. C. Luvizon and H. Tabia and D. Picard, “Human pose regression by combining indirect part detection and contextual information,” 2017. 3

[32] A. Nibali, Z. He, S. Morgan, L. Prendergast, “Numerical Coordinate Regression with Convolutional Neural Networks,” 2018. 3

[33] L. Ellenberger, F. Oberle, S. Lorenzetti, W. O. Frey, J. G. Snedeker and J. Spörri, “Dynamic knee valgus in competitive alpine skiers: Observation from youth to elite and influence of biological maturation.” Scand J Med Sci Sports., 2020. 3

[34] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks..” In NIPS, 2015. 6

[35] D. Mehta, S. Sridhar, O. Sotnychenko , H. Rhodin, M. Shafiei, H. Seidel , W. Xu, D. Casas, C. Theobalt, “VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera.” In ACM Transactions on Graphics, Vol. 36, No. 4, Article 44, 2017. 6