2D/3D HUMAN POSE ESTIMATION USING DEEP CONVOLUTIONAL NEURAL NETS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUHAMMED KOCABAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2019

Approval of the thesis:

2D/3D HUMAN POSE ESTIMATION USING DEEP CONVOLUTIONAL NEURAL NETS

submitted by **MUHAMMED KOCABAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assist. Prof. Dr. Emre Akbaş
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assist. Prof. Dr. Gökberk Cinbiş
Computer Engineering, METU _____

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU _____

Assist. Prof. Dr. Hamdi Dibeklioğlu
Computer Engineering, Bilkent University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Muhammed Kocabaş

Signature :

ABSTRACT

2D/3D HUMAN POSE ESTIMATION USING DEEP CONVOLUTIONAL NEURAL NETS

Kocabaş, Muhammed
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Emre Akbaş

January 2019, 63 pages

In this thesis, we propose algorithms to estimate 2D/3D human pose from single view images.

In the first part of the thesis, we present *MultiPoseNet*, a novel bottom-up multi-person pose estimation architecture that combines a multi-task model with a novel assignment method. MultiPoseNet can jointly handle person detection, keypoint detection, person segmentation and pose estimation problems. The novel assignment method is implemented by the *Pose Residual Network (PRN)* which receives keypoint and person detections, and produces accurate poses by assigning keypoints to person instances. On the COCO keypoints dataset, our pose estimation method outperforms all previous bottom-up methods both in accuracy (+4-point mAP over previous best result) and speed; it also performs on par with the best top-down methods while being at least 4x faster. Our method is the fastest real time system with $\sim$23 frames/sec.

In the second part of the thesis, we present *EpipolarPose* which is a self-supervised training methodology for single person monocular human pose estimation and *Pose*

Structure Score, a structure aware performance measure for 3D human pose estimation. Training accurate 3D human pose estimators requires large amount of 3D ground-truth data which is costly to collect. Various weakly or self supervised pose estimation methods have been proposed due to lack of 3D data. Nevertheless, these methods, in addition to 2D ground-truth poses, require either additional supervision in various forms (*e.g.* unpaired 3D ground truth data, a small subset of labels) or the camera parameters in multiview settings. To address these problems, we present EpipolarPose, a self-supervised learning method for 3D human pose estimation, which does not need any 3D ground-truth data or camera extrinsics. During training, EpipolarPose estimates 2D poses from multi-view images, and then, utilizes epipolar geometry to obtain a 3D pose and camera geometry which are subsequently used to train a 3D pose estimator. We demonstrate the effectiveness of our approach on standard benchmark datasets *i.e.* Human3.6M and MPI-INF-3DHP where we set the new state-of-the-art among weakly/self-supervised methods. Furthermore, we propose a new performance measure Pose Structure Score (PSS) which is a scale invariant, structure aware measure to evaluate the structural plausibility of a pose with respect to its ground truth.

Keywords: Convolutional Neural Networks, Human Pose Estimation, Multi-view Geometry

ÖZ

DERİN EVRİŞİMSEL SİNİR AĞLARI İLE 2B/3B İNSAN VÜCUDU POZİSYON KESTİRİMİ

Kocabaş, Muhammed
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Emre Akbaş

Ocak 2019 , 63 sayfa

Bu tezde tekli görüntülerden 2B/3B insan pozisyon kestirimi için algoritmalar önerdik.

Tezin ilk kısmında, özgün bir atama tekniği ile çoklu-görev modelini birleştiren yeni bir aşağıdan-yukarıya çoklu insan pozisyon kestirimi algoritması olan *MultiPoseNet*'i önerdik. *MultiPoseNet* insan tespiti, ana nokta tespiti, insan bölütleme ve pozisyon kestirimi görevlerini beraber yürütebilmektedir. Yeni atama tekniği tespit edilen ana noktalar ve insanları eşleştirerek doğru pozlar üreten *Pose Residual Network (PRN)* ile gerçekleştirilmiştir. Geliştirdiğimiz poz kestirim yöntemi COCO ana nokta veri setinde tüm aşağıdan-yukarıya yöntemlerden hız (en iyi sonuçtan +4-puan mAP daha fazla) ve doğruluk bazında daha üstün sonuçlar vermektedir, ayrıca yukarıdan-aşağıya yöntemlerden 4 kat daha hızlı çalışırken doğruluk bazında onlara yakın sonuç üretebilmektedir. Yöntemimiz $\sim$23 çerçeve/saniye ile en hızlı çalışan gerçek zamanlı sistemdir.

Tezin ikinci kısmında, öz gözetimli tek insanlı monoküler görüntülerden 3B insan po-

zisyon kestirimi yöntemi olan *EpipolarPose*'u ve yapı farkındalıklı bir 3B insan pozisyon kestirimi performans ölçüsü olan *Pose Structure Score*'u önerdik. 3B insan pozisyon kestirimi yöntemlerini eğitmek elde etmesi oldukça maliyetli çok miktarda 3B gerçek referans etiketler gerektirmektedir. 3B verinin eksikliği nedeni ile birçok zayıf veya öz gözetimli poz kestirimi yöntemleri geliştirilmiştir. Buna rağmen bu yöntemler 2B gerçek referans etiketlerin yanında çeşitli şekillerde gözetim (örneğin eşleştirilmemiş 3B gerçek referans etiketler, etiketlerin ufak bir alt kümesi) ya da çoklu görüntü senaryolarında kamera değişkenlerine ihtiyaç duymaktadır. Bu problemleri çözmek amacıyla 3B gerçek referans etiketine veya kamera değişkenlerine ihtiyaç duymayan öz gözetimli öğrenme yöntemi olan *EpipolarPose*'u geliştirdik. Eğitim sırasında *EpipolarPose* bir 3B poz kestirim modelini eğitmek için çoklu görüntülerde 2B insan pozlarını tahmin eder, ardından epi-kutuplu geometri ile 3B pozu ve kamera geometrisini etiket olarak kullanır. Yaklaşımımızın etkisini Human3.6M ve MPI-INF-3DHP denektaşlarında en gelişmiş sonuçları elde ederek gösterdik. Ek olarak bir pozun gerçek referans değerine göre olan yapısal geçerliliğini ölçebilen, ölçekten bağımsız, yapı farkındalıklı yeni bir performans ölçüsü Pose Structure Score (PSS) önerdik.

Anahtar Kelimeler: Evrişimsel Sinir Ağları, İnsan Poz Kestirimi, Çok Yönlü Geometri

To my brothers in arms and our families

ACKNOWLEDGMENTS

I would like to express my deepest appreciation and gratitude to Asst. Prof. Emre Akbaş for all his help, guidance and supervision. This work would not have been possible without his continuous support and motivation.

I would like to thank my thesis committee members, Asst. Prof. Gökberk Cinbiş, and Asst. Prof. Hamdi Dibeklioğlu for their valuable feedback.

I would like to thank ImageLab members, particularly Salih Karagöz, for all their help and contribution to this work.

I would like to express my gratitude to my family and friends for their continuous support.

I would like to thank all of my teachers and commanders in Işıklar Askeri Hava Lisesi and Hava Harp Okulu for nurturing me along the way.

I would like to dedicate this thesis to my innocent brothers in arms who are punished with a life imprisonment.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
MLP	Multi Layer Perceptron
mAP	Mean Average Precision
FPS	Frame per Second
PRN	Pose Residual Network
SHG	Stacked Hourglass Networks
FPN	Feature Pyramid Networks
R-CNN	Regional CNN
OKS	Object Keypoint Similarity
COCO	Common Objects in Context
NMS	Non Maximum Suppression
MPJPE	Mean Per Joint Position Error
PCK	Percentage of Correct Keypoints
PSS	Pose Structure Score
H36M	Human 3.6 Million Dataset
3DHP	MPII 3D Human Pose Estimation Dataset
NMS	Non Maximum Suppression

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

This thesis is aimed at developing fast and accurate solutions for 2D and 3D human pose estimation from single view images. In the first part of the thesis, we try to solve multi person 2D pose estimation problem. In the second part, we focus on self supervised learning of 3D pose estimation from multi-view images. Here, we briefly analyze these problems. First part of the thesis is aimed at estimating the two-dimensional (2D) poses of multiple people in a given image. Any solution to this problem has to tackle a few sub-problems: detecting body joints (or keypoints, as they are called in the influential COCO [2] dataset) such as wrists, ankles, etc., grouping these joints into person instances, or detecting people and assigning joints to person instances. Depending on which sub-problem is tackled first, there have been two major approaches in multi-person 2D estimation: *bottom-up* and *top-down*. Bottom-up methods [3, 4, 5, 6, 7, 8, 9] first detect body joints without having any knowledge as to the number of people or their locations. Next, detected joints are grouped to form individual poses for person instances. On the other hand, top-down methods [10, 11, 12, 13] start by detecting people first and then for each person detection, a single-person pose estimation method (e.g. [14, 15, 16, 17]) is executed. Single-person pose estimation, i.e. detecting body joints conditioned on the information that there is a single person in the given input (the top-down approach), is typically a more costly process than grouping the detected joints (the bottom-up approach). Consequently, the top-down methods tend to be slower than the bottom-up methods, since they need to repeat the single-person pose estimation for each person detection; however, they usually yield better accuracy than bottom-up methods.

In the second part, we study the self supervised learning of 3D human pose estimation. Human pose estimation in the wild is a challenging problem in computer vision. Although there are large-scale datasets [18, 19] for two-dimensional (2D) pose estimation, 3D datasets [20, 21] are either limited to laboratory settings or limited in size and diversity. Since collecting 3D human pose annotations in the wild is costly and 3D datasets are limited, researchers have resorted to weakly or self supervised approaches with the goal of obtaining an accurate 3D pose estimator by using minimal amount of additional supervision on top of the existing 2D pose datasets. Various methods have been developed to this end. These methods, in addition to ground-truth 2D poses, require either additional supervision in various forms (such as unpaired 3D ground truth data[22], a small subset of labels [23]) or (extrinsic) camera parameters in multiview settings [24]. To the best of our knowledge, there is only one method [25] which can produce a 3D pose estimator by using only 2D ground-truth. In this paper, we propose another such method.

1.2 Proposed Methods and Models

First, we present a new bottom-up method for multi-person 2D pose estimation. Our method is based on a multi-task learning model which can jointly handle the person detection, keypoint detection, person segmentation and pose estimation problems. To emphasize its multi-person and multi-task aspects of our model, we named it as "MultiPoseNet." Our model (Fig. 2.1) consists of a shared backbone for feature extraction, detection subnets for keypoint and person detection/segmentation, and a final network which carries out the pose estimation, i.e. assigning detected keypoints to person instances.

Our major contribution lies in the pose estimation step where the network implements a novel assignment method. This network receives keypoint and person detections, and produces a pose for each detected person by assigning keypoints to person boxes using a learned function. In order to put our contribution into context, here we briefly describe the relevant aspects of the state-of-the-art (SOTA) bottom-up methods [3, 9]. These methods attempt to group detected keypoints by exploiting lower order relations either between the group and keypoints, or among the keypoints themselves.

Specifically, Cao et al. [3] model pairwise relations (called part affinity fields) between two nearby joints and the grouping is achieved by propagating these pairwise affinities. In the other SOTA method, Newell et al. [9] predict a real number called a *tag* per detected keypoint, in order to identify the group the detection belongs to. Hence, this model makes use of the unary relations between a certain keypoint and the group it belongs to. Our method generalizes these two approaches in the sense that we achieve the grouping in a single shot by considering all joints together at the same time. We name this part of our model which achieves the grouping as the *Pose Residual Network* (PRN) (Fig. 2.2). PRN takes a region-of-interest (RoI) pooled keypoint detections and then feeds them into a residual multilayer perceptron (MLP). PRN considers all joints simultaneously and learns configurations of joints. We illustrate this capability of PRN by plotting a sample set of learned configurations. (Fig. 2.2 right).

Our experiments (on the COCO dataset, using no external data) show that our method outperforms all previous bottom-up methods: we achieve a 4-point mAP increase over the previous best result. Our method performs on par with the best performing top-down methods while being an order of magnitude faster than them. To the best of our knowledge, there are only two top-down methods that we could not outperform. Given the fact that bottom-up methods have always performed less accurately than the top-down methods, our results are remarkable.

In terms of running time, our method appears to be the fastest of all multi-person 2D pose estimation methods. Depending on the number of people in the input image, our method runs at between 27 frames/sec (FPS) (for one person detection) and 15 FPS (for 20 person detections). For a typical COCO image, which contains $\sim$3 people on average, we achieve $\sim$23 FPS (Fig. 2.8).

Second, we present a new way to train 3D pose estimation models without 3D supervision. Our method, "EpiloparPose," uses 2D pose estimation and epipolar geometry to obtain 3D poses, which are subsequently used to train a 3D pose estimator. EpipolarPose works with an arbitrary number of cameras (must be at least 2) and it does not need any 3D supervision or the extrinsic camera parameters, however, it can utilize them if provided. On the Human3.6M [20] and MPI-INF-3DHP [21] datasets, we set

the new state-of-the-art in 3D pose estimation for weakly/self-supervised methods.

Human pose estimation allows for subsequent higher level reasoning, *e.g.* in autonomous systems (cars, industrial robots) and activity recognition. In such tasks, structural errors in pose might be more important than the localization error measured by the traditional evaluation metrics such as MPJPE (mean per joint position error) and PCK (percentage of correct keypoints). These metrics treat each joint independently, hence, fail to asses the whole pose as a structure. Figure 3.4 shows that structurally very different poses yield the same MPJPE with respect to a reference pose. To address this issue, we propose a new performance measure, called the Pose Structure Score (PSS), which is sensitive to structural errors in pose. PSS computes a scale invariant performance score with the capability to score the structural plausibility of a pose with respect to its ground truth. Note that PSS is not a loss function, it is a performance measure that can be used along with MPJPE and PCK to describe the representation capacity of a pose estimator.

To compute PSS, we first need to model the natural distribution of ground-truth poses. To this end, we use an unsupervised clustering method. Let $\mathbf{p}$ be the predicted pose for an image whose ground-truth is $\mathbf{q}$. First, we find which cluster centers are closest to $\mathbf{p}$ and $\mathbf{q}$. If both of them are closest to (*i.e.* assigned to) the same cluster, then the pose structure score (PSS) of $\mathbf{q}$ is said to be 1, otherwise it is 0.

1.3 Contributions and Novelties

Our contributions are as follows:

- We propose the *Pose Residual Network* (PRN), a simple yet very effective method for the problem of assigning/grouping body joints.
- We outperform all previous bottom-up methods and achieve comparable performance with top-down methods.
- Our method works faster than all previous methods, in real-time at $\sim$23 frames/sec.
- Our network architecture is extendible; we show that using the same backbone, one can solve other related problems, too, e.g. person segmentation.

- We present EpipolarPose, a method that can predict 3D human pose from a single-image. For training, EpipolarPose does not require any 3D supervision nor camera extrinsics. It creates its own 3D supervision by utilizing epipolar geometry and 2D ground-truth poses.
- We set the new state-of-the-art among weakly/self-supervised methods for 3D human pose estimation.
- We present Pose Structure Score (PSS), a new performance measure for 3D human pose estimation to better capture structural errors.

The work presented in this thesis has appeared in the following papers:

- Kocabas, M., Karagoz, S., & Akbas, E. Self Supervised Learning of Human Pose Estimation using Multiple View Geometry. Submitted to Conference on Computer Vision and Pattern Recognition, 2019.
- Kocabas, M., Karagoz, S., & Akbas, E. MultiPoseNet: Fast Multi-Person Pose Estimation using Pose Residual Network. European Conference on Computer Vision, 2018.

1.4 The Outline of the Thesis

Chapter 2 introduces the MultiPoseNet model along with our novel assignment method Pose Residual Network. We discuss the previous work done in the area of 2D human pose estimation. We present the detailed experiments and comparison to previous state-of-the-art methods in this section.

Chapter 3 describes the EpipolarPose and Pose Structure Score. The details about the self supervised training methodology for EpipolarPose are provided in that chapter. We also present detailed analysis with ablation experiments.

Chapter 4 provides a brief summary of the entire thesis, it also presents potential directions for future work that move forward with the proposed models in chapters 2 and 3.

CHAPTER 2

MULTIPOSENET: FAST MULTI-PERSON POSE ESTIMATION USING POSE RESIDUAL NETWORK

2.1 Introduction

This work is aimed at estimating the two-dimensional (2D) poses of multiple people in a given image. Any solution to this problem has to tackle a few sub-problems: detecting body joints (or keypoints[1], as they are called in the influential COCO [2] dataset) such as wrists, ankles, etc., grouping these joints into person instances, or detecting people and assigning joints to person instances. Depending on which sub-problem is tackled first, there have been two major approaches in multi-person 2D estimation: *bottom-up* and *top-down*. Bottom-up methods [3, 4, 5, 6, 7, 8, 9] first detect body joints without having any knowledge as to the number of people or their locations. Next, detected joints are grouped to form individual poses for person instances. On the other hand, top-down methods [10, 11, 12, 13] start by detecting people first and then for each person detection, a single-person pose estimation method (e.g. [14, 15, 16, 17]) is executed. Single-person pose estimation, i.e. detecting body joints conditioned on the information that there is a single person in the given input (the top-down approach), is typically a more costly process than grouping the detected joints (the bottom-up approach). Consequently, the top-down methods tend to be slower than the bottom-up methods, since they need to repeat the single-person pose estimation for each person detection; however, they usually yield better accuracy than bottom-up methods.

In this work, we present a new bottom-up method for multi-person 2D pose estima-

[1] We use "body joint" and "keypoint" interchangeably throughout the thesis.

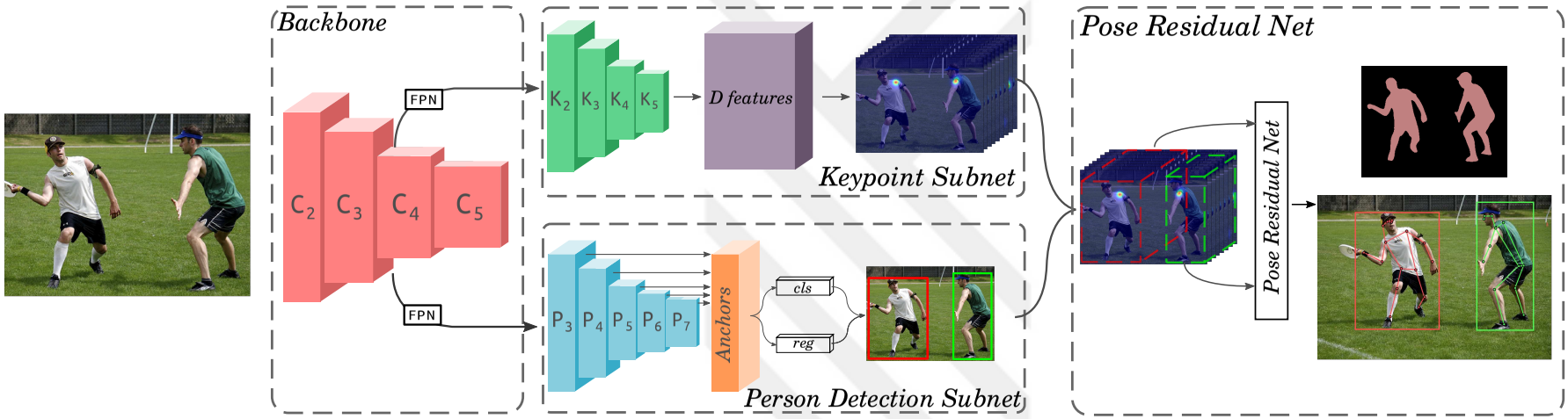

Figure 2.1: MultiPoseNet is a multi-task learning architecture capable of performing human keypoint estimation, detection and semantic segmentation tasks altogether efficiently.

tion. Our method is based on a multi-task learning model which can jointly handle the person detection, keypoint detection, person segmentation and pose estimation problems. To emphasize its multi-person and multi-task aspects of our model, we named it as "MultiPoseNet." Our model (Fig. 2.1) consists of a shared backbone for feature extraction, detection subnets for keypoint and person detection/segmentation, and a final network which carries out the pose estimation, i.e. assigning detected keypoints to person instances.

Our major contribution lies in the pose estimation step where the network implements a novel assignment method. This network receives keypoint and person detections, and produces a pose for each detected person by assigning keypoints to person boxes using a learned function. In order to put our contribution into context, here we briefly describe the relevant aspects of the state-of-the-art (SOTA) bottom-up methods [3, 9]. These methods attempt to group detected keypoints by exploiting lower order relations either between the group and keypoints, or among the keypoints themselves. Specifically, Cao et al. [3] model pairwise relations (called part affinity fields) between two nearby joints and the grouping is achieved by propagating these pairwise affinities. In the other SOTA method, Newell et al. [9] predict a real number called a *tag* per detected keypoint, in order to identify the group the detection belongs to. Hence, this model makes use of the unary relations between a certain keypoint and the group it belongs to. Our method generalizes these two approaches in the sense that we achieve the grouping in a single shot by considering all joints together at the same time. We name this part of our model which achieves the grouping as the *Pose Residual Network* (PRN) (Fig. 2.2). PRN takes a region-of-interest (RoI) pooled keypoint detections and then feeds them into a residual multilayer perceptron (MLP). PRN considers all joints simultaneously and learns configurations of joints. We illustrate this capability of PRN by plotting a sample set of learned configurations. (Fig. 2.2 right).

Our experiments (on the COCO dataset, using no external data) show that our method outperforms all previous bottom-up methods: we achieve a 4-point mAP increase over the previous best result. Our method performs on par with the best performing top-down methods while being an order of magnitude faster than them. To the best of our knowledge, there are only two top-down methods that we could not outperform.

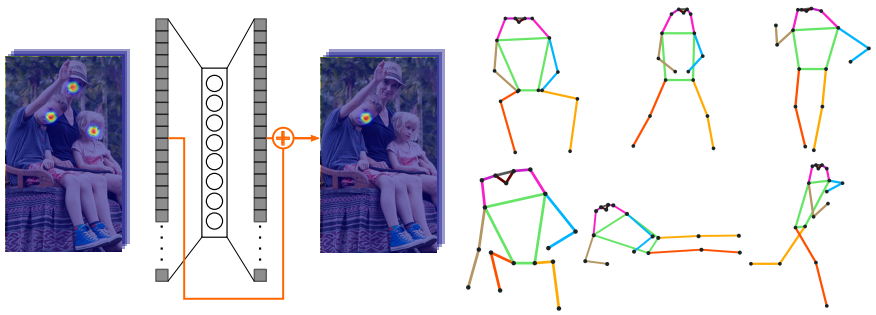

Figure 2.2: **Left:** Pose Residual Network (PRN). The PRN is able to disambiguate which keypoint should be assigned to the current person box. **Right:** Six sample poses obtained via clustering the structures learned by PRN.

Given the fact that bottom-up methods have always performed less accurately than the top-down methods, our results are remarkable.

In terms of running time, our method appears to be the fastest of all multi-person 2D pose estimation methods. Depending on the number of people in the input image, our method runs at between 27 frames/sec (FPS) (for one person detection) and 15 FPS (for 20 person detections). For a typical COCO image, which contains $\sim$3 people on average, we achieve $\sim$23 FPS (Fig. 2.8).

Our contributions in this work are four fold. (1) We propose the *Pose Residual Network* (PRN), a simple yet very effective method for the problem of assigning/grouping body joints. (2) We outperform all previous bottom-up methods and achieve comparable performance with top-down methods. (3) Our method works faster than all previous methods, in real-time at $\sim$23 frames/sec. (4) Our network architecture is extendible; we show that using the same backbone, one can solve other related problems, too, e.g. person segmentation.

2.2 Related Work

2.2.1 Single Person Pose Estimation

Single person pose estimation is to predict individual body parts given a cropped person image (or, equivalently, given its exact location and scale within an image).

Early methods (prior to deep learning) used hand-crafted HOG features [26] to detect body parts and probabilistic graphical models to represent the pose structure (tree-based [27, 28, 29, 30]; non-tree based [31, 32]).

Deep neural networks based models [33, 34, 35, 14, 15, 36, 28, 37, 17, 38] have quickly dominated the pose estimation problem after the initial work by Toshev et al. [33] who used the AlexNet architecture to directly regress spatial joint coordinates. Tompson et al. [34] learned pose structure by combining deep features along with graphical models. Carreira et al. [35] proposed the Iterative Error Feedback method to train Convolutional Neural Networks (CNNs) where the input is repeatedly fed to the network along with current predictions in order to refine the predictions. Wei et al.[14] were inspired by the pose machines [39] and used CNNs as feature extractors in pose machines. *Hourglass blocks*, (HG) developed by Newell et al. [15], are basically convolution-deconvolution structures with residual connections. Newell et al. stacked HG blocks to obtain an iterative refinement process and showed its effectiveness on single person pose estimation. Stacked Hourglass (SHG) based methods made a remarkable performance increase over previous results. Chu et al. [36] proposed adding visual attention units to focus on keypoint regions of interest. Pyramid residual modules by Yang et al.[28] improved the SHG architecture to handle scale variations. Lifshitz et al. [37] used a probabilistic keypoint voting scheme from image locations to obtain agreement maps for each body part. Belagiannis et al. [38] introduced a simple recurrent neural network based prediction refinement architecture. Huang et al.[17] developed a coarse-to-fine model with Inception-v2 [40] network as the backbone. The authors calculated the loss in each level of the network to learn coarser to finer representations of parts.

2.2.2 Multi Person Pose Estimation

2.2.2.1 Bottom-up

Multi person pose estimation solutions branched out as bottom-up and top-down methods. Bottom-up approaches detect body joints and assign them to people instances, therefore they are faster in test time and smaller in size compared to top-

down approaches. However, they miss the opportunity to zoom into the details of each person instance. This creates an accuracy gap between top-down and bottom-up approaches.

In an earlier work by Ladicky et al.[41], they proposed an algorithm to jointly predict human part segmentations and part locations using HOG-based features and probabilistic approach. Gkioxari et al. [42] proposed k-poselets to jointly detect people and keypoints.

Most of the recent approaches use Convolutional Neural Networks (CNNs) to detect body parts and relationships between them in an end-to-end manner [3, 9, 43, 4, 27, 5], then use assignment algorithms [3, 4, 5, 43] to form individual skeletons.

Pischulin et al.[4] used deep features for joint prediction of part locations and relations between them, then performed correlation clustering. Even though [4] doesn't use person detections, it is very slow due to proposed clustering algorithm and processing time is in the order of hours. In a following work by Insafutdinov et al.[5], they benefit from deeper ResNet architectures as part detectors and improved the parsing efficiency of a previous approach with an incremental optimization strategy. Different from Pischulin and Insafutdinov, Iqbal et al. [44] proposed to solve the densely connected graphical model locally, thus improved time efficiency significantly.

Cao et al.[3] built a model that contain two entangled CPM[14] branches to predict keypoint heatmaps and pairwise relationships (part affinity fields) between them. Keypoints are grouped together with fast Hungarian bipartite matching algorithm according to conformity of part affinity fields between them. This model runs in realtime. Newell et al.[9] extended their SHG idea by outputting associative vector embeddings which can be thought as tags representing each keypoint's group. They group keypoints with similar tags into individual people.

2.2.2.2 Top-down

Top-down methods first detect people (typically using a top performing, off-the-shelf object detector) and then run a single person pose estimation (SPPEN) method per person to get the final pose predictions. Since a SPPEN model is run for each person

instance, top-down methods are extremely slow, however, each pose estimator can focus on an instance and perform fine localization. Papandreou et al.[11] used ResNet with dilated convolutions [45] which has been very successful in semantic segmentation [46] and computing keypoint heatmap and offset outputs. In contrast to Gaussian heatmaps, the authors estimated a disk-shaped keypoint masks and 2-D offset vector fields to accurately localize keypoints. Joint part segmentation and keypoint detection given human detections approach were proposed by Xia et al.[47] The authors used separate PoseFCN and PartFCN to obtain both part masks and locations and fused them with fully-connected CRFs. This provides more consistent predictions by eliminating irrelevant detections. Fang et al.[13] proposed to use spatial transformer networks to handle inaccurate bounding boxes and used *stacked hourglass* blocks [15]. He et al.[12] combined instance segmentation and keypoint prediction in their *Mask-RCNN* model. They append keypoint heads on top of *RoI aligned* feature maps to get a one-hot mask for each keypoint. Chen et al.[10] developed *globalnet* on top of *Feature Pyramid Networks* [48] for multiscale inference and refined the predictions by using hyper-features [49].

2.3 The Method and Models

The architecture of our proposel model, MultiPoseNet, can be found in Fig. 2.1. In the following, we describe each component in detail.

2.3.1 The Shared Backbone

The backbone of MultiPoseNet serves as a feature extractor for keypoint and person detection subnets. It is actually a ResNet [45] with two Feature Pyramid Networks (FPN)[48] (one for the keypoint subnet, the other for the person detection subnet) connected to it, FPN creates pyramidal feature maps with top-down connections from all levels of CNN's feature hierarchy to make use of inherent multi-scale representations of a CNN feature extractor. By doing so, FPN compromises high resolution, weak representations with low resolution, strong representations. Powerful localization and classification properties of FPN proved to be very successful in detection,

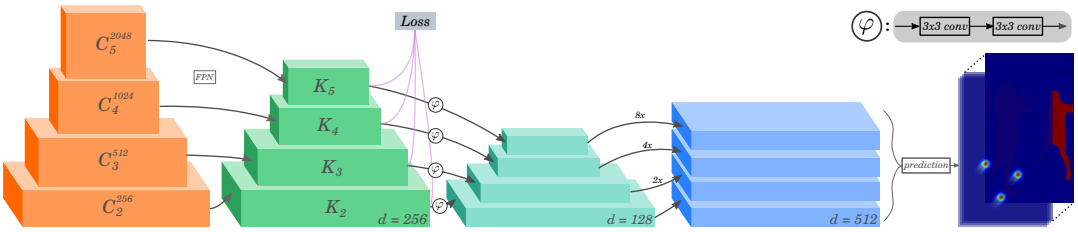

Figure 2.3: The architecture of the keypoint subnet. It takes hierarchical CNN features as input and outputs keypoint and segmentation heatmaps.

segmentation and keypoint tasks recently [10, 12, 50, 48]. In our model, we extracted features from the last residual blocks C_2, C_3, C_4, C_5 with strides of (4,8,16,32) pixels and compute corresponding FPN features per subnet.

2.3.2 Keypoint Estimation Subnet

Keypoint estimation subnet (Fig. 2.3) takes hierarchical CNN features (outputted by the corresponding FPN) and outputs keypoint and segmentation heatmaps. Heatmaps represent keypoint locations as Gaussian peaks. Each heatmap layer belongs to a specific keypoint class (nose, wrists, ankles etc.) and contains arbitrary number of peaks that pertain to person instances. Person segmentation mask at the last layer of heatmaps encodes the pixelwise spatial layout of people in the image.

A set of features specific to the keypoint detection task are computed similarly to [48] with top-down and lateral connections from the bottom-up pathway. $K_2 - K_5$ features have the same spatial size corresponding to $C_2 - C_5$ blocks but the depth is reduced to 256. K features are identical to P features in the original FPN paper, but we denote them with K to distinguish from person detection subnet layers. The depth of P features is downsized to 128 with 2 subsequent 3×3 convolutions to obtain D_2, D_3, D_4, D_5 layers. Since D features still have different strides, we upsampled D_3, D_4, D_5 accordingly to match 4-pixel stride as D_2 features and concatenated them into a single depth-512 feature map. Concatenated features are smoothed by a 3×3 convolution with ReLU. Final heatmap which has $(K + 1)$ layers obtained via 1×1 convolutions without activation. The final output is multiplied with a binary mask of $\mathbf{W}$ which has $\mathbf{W}(\mathbf{p}) = 0$ in the area of the persons without annotation. K is

(a) (b) (c) (d)

Figure 2.4: Bounding box overlapping scenarios.

the number of human keypoints annotated in a dataset and $+1$ is person segmentation mask. In addition to the loss applied in the last layer, we append a loss at each level of K features to benefit from intermediate supervision. Semantic person segmentation masks are predicted in the same way with keypoints.

2.3.3 Person Detection Subnet

Modern object detectors are classified as one-stage (SSD[51], YOLO[52], RetinaNet [50]) or two-stage (Fast R-CNN[53], Faster R-CNN[54]) detectors. One-stage detectors enable faster inference but have lower accuracy in comparison to two-stage detectors due to foreground-background class imbalance. The recently proposed RetinaNet [50] model improved one-stage detectors' performance with *focal loss* which can handle the class imbalance problem during training. In order to design a faster and simpler person detection model which is compatible with FPN backbone, we have adopted RetinaNet. Same strategies to compute anchors, losses and pyramidal image features are followed. Classification and regression heads are modified to handle only person annotations.

2.3.4 Pose Residual Network (PRN)

Assigning keypoint detections to person instances (bounding boxes, in our case) is straightforward if there is only one person in the bounding box as in Fig. 2.4 a-b. However, it becomes non-trivial if there are overlapping people in a single box as in Fig. 2.4 c-d. In the case of an overlap, a bounding box can contain multiple keypoints not related to the person in question, so this creates ambiguity in constructing final

pose predictions. We solve these ambiguities by learning pose structures from data.

The input to PRN is prepared as follows. For each person box that the person detection subnet detected, the region from the keypoint detection subnet's output, corresponding to the box, is cropped and resized to a fixed size, which ensures that PRN can handle person detections of arbitrary sizes and shapes. Specifically, let $\mathbf{X}$ denote the input to the PRN, where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ in which $\mathbf{x}_k \in \mathbb{R}^{W \times H}$, k is the number of different keypoint types. The final goal of PRN is to output $\mathbf{Y}$ where $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$, in which $\mathbf{y}_k \in \mathbb{R}^{W \times H}$ is of the same size as $\mathbf{x}_k$, containing the correct position for each keypoint indicated by a peak in that keypoint's channel. PRN models the mapping from $\mathbf{X}$ to $\mathbf{Y}$ as

$$\mathbf{y}_k = \phi_k(\mathbf{X}) + \mathbf{x}_k \tag{21}$$

where the functions $\phi_1(\cdot), \ldots, \phi_K(\cdot)$ apply a *residual correction* to the pose in $\mathbf{X}$, hence the name pose residual network. We implement Eq. 21 using a residual multilayer perceptron Fig. 2.2. Activation of the output layer uses softmax to obtain a proper probability distribution and binary cross-entropy loss is used during training.

Before we came up with this residual model, we experimented with two naive baselines and a non-residual model. In the first baseline method, which we call *Max*, for each keypoint channel k, we find the location with the highest value and place a Gaussian in the corresponding location of the k^{th} channel in $\mathbf{Y}$. In the second baseline method, we compute $\mathbf{Y}$ as

$$\mathbf{y}_k = \mathbf{x}_k * \mathbf{P}_k \tag{22}$$

where $\mathbf{P}_k$ is a prior map for the location of the k^{th} joint, learned from ground-truth data and $*$ is element-wise multiplication. We named this method as Unary Conditional Relationship (UCR). Finally, in our non-residual model, we implemented

$$\mathbf{y}_k = \phi_k(\mathbf{X}). \tag{23}$$

Performances of all these models can be found in Table 2.3.

In the context of the models described above, both SOTA bottom up methods learn lower order grouping models than the PRN. Cao et al. [3] model pairwise channels in $\mathbf{X}$ while Newell et al. [9] model only unary channels in $\mathbf{X}$. Hence, our model can be considered as a generalization of these lower order grouping models.

We hypothesize that each node in PRN's hidden layer encodes a certain body configuration. To show this, we visualized some of the representative outputs of PRN in Fig. 2.2. These poses is obtained via reshaping PRN outputs and selecting the maximum activated keypoints to form skeletons. All obtained configurations are clustered using k-means with OKS (object keypoint similarity)[2] and cluster means are visualized in Fig. 2.2. OKS (object keypoint similarity) is used as k-means distance metric to cluster the meaningful poses.

2.3.5 Implementation Details

2.3.5.1 Training

Due to different convergence times and loss imbalance, we have trained keypoint and person detection tasks separately. To use the same backbone in both task, we first trained the model with only keypoint subnet Fig. 2.3. Thereafter, we froze the backbone parameters and trained the person detection subnet. Since the two tasks are semantically similar, person detection results were not adversely affected by the frozen backbone.

We have utilized Tensorflow [55] and Keras [56] deep learning library to implement training and testing procedures. For person detection, we made use of open-source Keras RetinaNet[57] implementation.

Keypoint Estimation Subnet: For keypoint training, we used 480x480 image patches, that are centered around the crowd or the main person in the scene. Random rotations between ± 40 degrees, random scaling between $0.8 - 1.2$ and vertical flipping with a probability of 0.3 was used during training. We have transferred the ImageNet [58]

pretrained weights for each backbone before training. We optimize the model with Adam [59] starting from learning rate 1e-4 and decreased it by a factor of 0.1 in plateaux. We used the Gaussian peaks located at the keypoint locations as the ground truth to calculate L_2 loss, and we masked (ignored) people that are not annotated. We appended the segmentation masks to ground-truth as an extra layer and trained along with keypoint heatmaps. The cost function that we minimize is

$$L_{kp} = \mathbf{W} \cdot \|\mathbf{H}_t - \mathbf{H}_p\|_2^2 , \tag{24}$$

where $\mathbf{H}_t$ and $\mathbf{H}_p$ are the ground-truth and predicted heatmaps respectively, and $\mathbf{W}$ is the mask used to ignore non-annotated person instances.

Person Detection Subnet: We followed a similar person detection training strategy as [50]. Images containing persons are used, they are resized such that shorter edge is 800 pixels. We froze backbone weights after keypoint training and not updated during person detection training. We optimized subnet with Adam [59] starting from learning rate 1e-5 and is decreased by a factor of 0.1 in plateaux. We used Focal loss with $(\gamma = 2, \alpha = 0.25)$ and smooth L_1 loss for classification and bbox regression, respectively. We obtained final proposals using NMS with a threshold of 0.3.

Pose Residual Network: During training, we cropped input and output pairs and resized heatmaps according to bounding-box proposals. All crops are resized to a fixed size of 36×56 (height/width = 1.56). We trained the PRN network separately and Adam optimizer [59] with a learning rate of 1e-4 is used during training. Since the model is shallow, convergence takes 1.5 hours approximately.

We trained the model with the person instances which has more than 2 keypoints. We utilized a sort of curriculum learning [60] by sorting annotations based on number of keypoints and bounding box areas. In each epoch, model is started to learn easy-to-predict instances, hard examples are given in later stages.

2.3.5.2 Inference

The whole architecture (see in Fig. 2.1) behaves as a monolithic, end-to-end model during test time. First, an image $(W \times H \times 3)$ is processed through backbone model to extract the features in multi-scales. Person and keypoint detection subnets compute outputs simultaneously out of extracted features. Keypoints are outputted as $W \times H \times (K+1)$ sized heatmaps. K is the number of keypoint channels, and $+1$ is for the segmentation channel. Person detections are in the form of $N \times 5$, where N is the number of people and 5 channel corresponds to 4 bounding box coordinates along with confidence scores. Keypoint heatmaps are cropped and resized to form RoIs according to person detections. Optimal RoI size is determined as $36 \times 56 \times (K+1)$ in our experiments. PRN takes each RoI as separate input, then outputs same size RoI with only one keypoint selected in each layer of heatmap. All selected keypoints are grouped as a person instance.

2.4 Experiments

2.4.1 Datasets

We trained our keypoint and person detection models on COCO keypoints dataset [2] (without using any external/extra data) in our experiments. We used COCO for evaluating the keypoint and person detection, however, we used PASCAL VOC 2012[61] for evaluating person segmentation due to the lack of semantic segmentation annotations in COCO. Backbone models (ResNet-50 and ResNet-101) were pretrained on ImageNet and we finetuned with COCO-keypoints.

COCO train2017 split contains 64K images including 260K person instances which 150K of them have keypoint annotations. Keypoints of persons with small area are not annotated in COCO. We did ablation experiments on COCO val2017 split which contains 2693 images with person instances. We made comparison to previous methods on the test-dev2017 split which has 20K test images. We evaluated test-dev2017 results on the online COCO evaluation server. We use the official COCO evaluation metric average precision (AP) and average recall (AR). OKS and IoU based scores

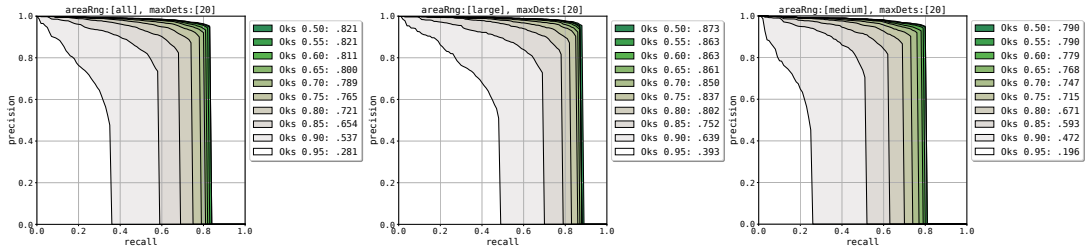

Figure 2.5: Precision-recall curves on COCO validation set across scales *all, large* and *medium*. Analysis tool is provided by [1]

were used for keypoint and person detection tasks, respectively.

We performed person segmentation evaluation in PASCAL VOC 2012 test split with PASCAL IoU metric. PASCAL VOC 2012 person segmentation test split contains 1456 images. We obtained "Test results" using the online evaluation server.

2.4.2 Multi Person Pose Estimation

We present the recall-precision curves of our method for different scales *all, large, medium* in Fig. 2.5. The overall AP results of our method along with top-performing bottom-up (BU) and top-down (TD) methods are given in Table 2.1. MultiPoseNet outperforms all bottom-up methods and most of the top-down methods. We outperform the previously best bottom-up method[9] by a 4-point increase in mAP. In addition, the runtime speed (see the FPS column Table 2.1 and Fig. 2.8) of our system is far better than previous methods with 23 FPS on average[2]. This proves the effectiveness of PRN for assignment and our multitask detection approach while providing reasonable speed-accuracy tradeoff. To get these results (Table 2.1) on test-dev, we have utilized test time augmentation and ensembling (as also done in all previous studies). Multi scale and multi crop testing was performed during test time data augmentation. Two different backbones and a single person pose refinement network similar to our keypoint detection model was used for ensembling. Results from dif-

[2] We obtained the FPS results by averaging the inference time using images containing 3 people (avg. number of person annotations per image in COCO dataset) on a GTX1080Ti GPU. Except for CFN and Mask RCNN, we obtained the FPS numbers by running the models ourselves under equal conditions. CFN's code is not available and Mask RCNN's code was only made recently available and we did not have time to test it ourselves. We got CFN's and Mask RCNN's FPS from their respective papers.

Table 2.1: Results on COCO **test-dev**, excluding systems trained with external data. Top-down methods are shown separately to make a clear comparison between bottom-up methods.

		FPS	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_{M}$**	**AP$_{L}$**	**AR**	**AR$_{50}$**	**AR$_{75}$**	**AR$_{M}$**	**AR$_{L}$**
BU	Ours	23	**69.6**	86.3	**76.6**	**65.0**	**76.3**	**73.5**	0.881	**79.5**	**68.6**	**80.3**
BU	Newell et al. [9]	6	65.5	**86.8**	72.3	60.6	72.6	70.2	**89.5**	76.0	64.6	78.1
BU	CMU-Pose [3]	10	61.8	84.9	67.5	57.1	68.2	66.5	87.2	71.8	60.6	74.6
TD	Megvii [10]	-	73.0	91.7	80.9	69.5	78.1	79.0	95.1	85.9	74.8	84.6
TD	CFN [17]	3	72.6	86.7	69.7	78.3	64.1	-	-	-	-	-
TD	Mask R-CNN [12]	5	69.2	90.4	76.0	64.9	76.3	75.2	93.7	81.1	70.3	81.8
TD	SJTU [13]	0.4	68.8	87.5	75.9	64.6	75.1	73.6	91.0	79.8	68.9	80.2
TD	GRMI-2017[3] [11]	-	66.9	86.4	73.6	64.0	72.0	71.6	89.2	77.6	66.1	79.1
TD	G-RMI-2016 [11]	-	60.5	82.2	66.2	57.6	66.6	66.2	86.6	71.4	61.9	72.2

ferent models are gathered and redundant detections was removed via OKS based NMS [11].

During ablation experiments we have inspected the effect of different backbones, keypoint detection architectures, and PRN designs. In Table 2.2 and 2.3 you can see the ablation analysis results on COCO validation set.

2.4.2.1 Different Backbones

We used ResNet models[45] as shared backbone to extract features. In Table 2.2, you can see the impact of deeper features and dilated features. R101 improves the result 1.6 mAP over R50. Dilated convolutions [46] which is very successful in dense detection tasks increases accuracy 2 mAP over R50 architecture. However, dilated convolutional filters add more computational complexity, consequently hinder real-time performance. We showed that concatenation of K features and intermediate supervision (see Section 2.3.2 for explanations) is crucial for good perfomance. The results demonstrated that performance of our system can be further enhanced with stronger feature extractors like recent ResNext [62] architectures.

Table 2.2: **Left:** Comparison of different keypoint models. **Right:** Performance of different backbone architectures. *(no concat: no concatenation, no int: no intermediate supervision, dil: dilated, concat: concatenation)*

Models	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_{M}$**	**AP$_{L}$**
R101$_{\text{no int.}}$	61.3	83.7	69.6	56.6	67.4
R101$_{\text{no concat}}$	62.1	84.3	70.9	57.3	68.8
R101	63.9	87.1	73.2	58.1	72.2
R101$_{\text{dil}}$	**64.3**	**88.2**	**75**	**59.6**	**73.9**

Backbones	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_{M}$**	**AP$_{L}$**
R50	62.3	86.2	71.9	57.7	70.4
R101	63.9	87.1	73.2	58.1	72.2
R101$_{\text{dil}}$	**64.3**	**88.2**	**75**	**59.6**	**73.9**

2.4.2.2 Different Keypoint Architectures

Keypoint estimation requires dense prediction over spatial locations, so its performance is dependent on input and output resolution. In our experiments, we used 480×480 images as inputs and outputted $120 \times 120 \times (K + 1)$ heatmaps per input. K is equal to 17 for COCO dataset. The lower resolutions harmed the mAP results while higher resolutions yielded longer training and inference complexity. We have listed the results of different keypoint models in Table 2.2.

The intermediate loss which is appended to the outputs of K block's enhanced the precision significantly. Intermediate supervision acts as a refinement process among the hierarchies of features. As previously shown in [3, 15, 14], it is an essential strategy in most of the dense detection tasks.

We have applied a final loss to the concatenated D features which is downsized from K features. This additional stage ensured us to combine multi-level features and compress them into a uniform space while extracting more semantic features. This strategy brought 2 mAP gain in our experiments.

2.4.2.3 Pose Residual Network Design

PRN is a simple yet effective assignment strategy, and is designed for faster inference while giving reasonable accuracy. To design an accurate model we have tried different configurations. Different PRN models and corresponding results can be seen in Table

Table 2.3: **Left:** Performance of different PRN models on COCO validation set. *N: nodes, D: dropout and R: residual connection.* **Right:** Ablation experiments of PRN with COCO validation data.

PRN Models	AP	AP_{50}	AP_{75}	AP_M	AP_L
1 Layer 50 N	76.3	89.2	79.1	74.8	80.4
1 Layer 50 N, D	78.6	91.7	82.4	77.1	83.1
1 Layer 512 N, D	84.1	94.2	85.3	82	86.2
2 Layers 512 N, D	81.9	91.1	82.6	79.8	84.3
1 Layer 2048 N, D+R	83.2	95.7	86.1	82.0	86.3
1 Layer 1024 N, D+R	**89.4**	**97.1**	**91.2**	**87.9**	**91.8**

PRN Ablations	AP	AP_{50}	AP_{75}	AP_M	AP_L
Both GT	89.4	97.1	91.2	87.9	91.8
GT keypoints + Our bbox	75.3	82.1	78	70.1	84.5
Our keypoints + GT bbox	65.1	89.2	76.2	60.3	74.7
PRN	64.3	88.2	75	59.6	73.9
UCR	49.7	59.5	52.4	44.1	51.6
Max	45.3	55.1	48.8	40.6	46.9

Table 2.4: PRN assignment results with non-grouped keypoints obtained from two bottom-up methods.

Models	AP	AP_{50}	AP_{75}	AP_M	AP_L
Cao et al. [3]	58.4	81.5	62.6	**54.4**	65.1
PRN + [3]	**59.2**	**82.2**	**64.4**	54.1	**67.0**
Newell et al. [9]	56.9	80.8	61.3	49.9	**68.8**
PRN + [9]	**58.1**	**81.4**	**63.0**	**51.3**	68.1

2.3. These results indicate the scores obtained from the assignment of ground truth person bounding boxes and keypoints.

We started with a primitive model which is a single hidden-layer MLP with 50 nodes, and added more nodes, regularization and different connection types to balance speed and accuracy. We found that 1024 nodes MLP, dropout with 0.5 probability and residual connection between input and output boosts the PRN performance up to 89.4 mAP on ground truth inputs.

In ablation analysis of PRN (see Table 2.3), we compared *Max*, *UCR* and *PRN* implementations (see Section 2.3.4 for descriptions) along with the performance of PRN with ground truth detections. We found that , lower order grouping methods could not handle overlapping detections, both of them performed poorly. As we hypothesized, PRN could overcome ambiguities by learning meaningful pose structures (Fig. 2.2 (right)) and improved the results by $\sim$20 mAP over naive assignment techniques. We evaluated the impact of keypoint and person subnets to the final results by al-

ternating inputs of PRN with ground truth detections. With ground truth keypoints and our person detections, we got 75.3 mAP, it shows that there is a large room for improvement in the keypoint localization part. With our keypoints and ground truth person detections, we got 65.1 mAP. This can be interpreted as our person detection subnet is performing quite well. Both ground truth detections got 89.4 mAP, which is a good indicator of PRN performance. In addition to these experiments, we tested PRN on the keypoints detected by previous SOTA bottom-up models [3, 9]. Consequently, PRN performed better grouping (see Table 2.4) than their methods: *Part Affinity Fields*[3] and *Associative Embedding*[9] by improving both detection results by $\sim$1 mAP. To obtain results in Table 2.4, we have used COCO val split, our person bounding box results and the keypoint results from the official source code of the papers. Note that running PRN on keypoints that were not generated by MultiPoseNet is unfair to PRN because it is trained with our detection architecture. Moreover original methods use image features for assignment coupled with their detection scheme, nonetheless, PRN is able to outperform the other grouping methods.

2.4.3 Person Detection

We trained the person detection subnet only on COCO person instances by freezing the backbone with keypoint detection parameters. The person category results of our network with different backbones can be seen in Table 2.5. We compared our results with the original methods that we adopt in our architecture. Our model with both ResNet-50 and ResNet-101 backends outperformed the original implementations. This is not a surprising result since our network is only dealing with a single class whereas the original implementations handle 80 object classes.

2.4.4 Person Segmentation

Person segmentation output is an additional layer appended to the keypoint outputs. We obtained the ground truth labels by combining person masks into single binary mask layer, and we jointly trained segmentation with keypoint task. Therefore, it adds a very small complexity to the model. Evaluation was performed on PASCAL

Table 2.5: **Left:** Person detection results on COCO dataset. **Right:**Person segmentation results on PASCAL VOC 2012 test split.

Person Detectors	$\mathbf{AP}$	$\mathbf{AP_{50}}$	$\mathbf{AP_{75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$
Ours - R101	**52.5**	**81.5**	**55.3**	**35.2**	**59**	71
Ours - R50	51.3	81.4	53.6	34.9	58	68.1
RetinaNet [50]	50.2	77.7	53.5	31.6	**59**	**71.5**
FPN [48]	47.5	78	50.7	28.6	55	67.4

Segmentation	**IoU**
DeepLab v3 [63]	**92.1**
DeepLab v2 [46]	87.4
SegNet [64]	74.9
Ours	87.8

VOC 2012 test set with PASCAL IoU metric. We obtained final segmentation results via multi-scale testing and thresholding. We did not apply any additional test-time augmentation or ensembling. Table 2.5 shows the test results of our system in comparison with previous successful semantic segmentation algorithms. Our model outperformed most of the successful baseline models such as SegNet [64] and Deeplab-v2 [46], and got comparable performance to the state-of-the-art Deeplab v3 [63] model. This demonstrates the capacity of our model to handle different tasks altogether with competitive performance. Some qualitative segmentation results are given in Fig. 2.6.

2.4.5 Runtime Analysis

Our system consists of a backbone, keypoint & person detection subnets, and the pose residual network. The parameter sizes of each block is given in Fig. 2.7. Most of the parameters are required to extract features in the backbone network, subnets and PRN are relatively lightweight networks. So most of the computation time is spent on the feature extraction stage. By using a shallow feature extractor like ResNet-50, we can achieve realtime performance. To measure the performance, we have built a model using ResNet-50 with 384×576 sized inputs which contain 1 to 20 people. We measured the time spent during the inference of 1000 images, and averaged the inference times to get a consistent result (see Fig. 2.8). Keypoint and person detections take 35 ms while PRN takes 2 ms per instance. So, our model can perform between 27 (1 person) and 15 (20 persons) FPS depending on the number of people.

Figure 2.6: Some qualitative results for COCO test-dev dataset.

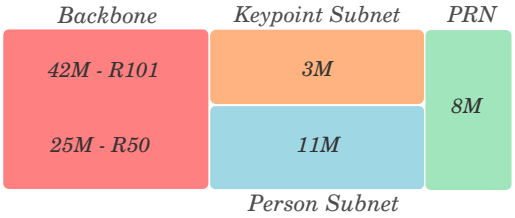

Figure 2.7: Number of parameters for each block of MultiPoseNet.

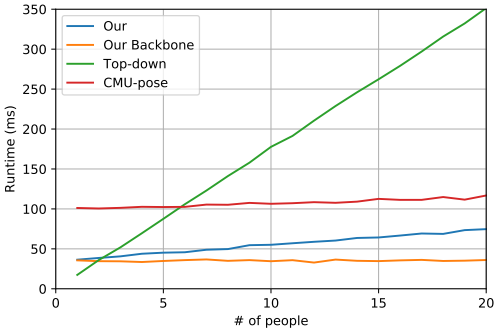

Figure 2.8: Runtime analysis of MultiPoseNet with respect to number of people.

2.5 Conclusion

In this work, we introduced the Pose Residual Network that is able to accurately assign keypoints to person detections outputted by a multi task learning architecture (MultiPoseNet). Our pose estimation method achieved state-of-the-art performance among bottom-up methods and comparable results with top-down methods. Our method has the fastest inference time compared to previous methods. We showed the assignment performance of pose residual network ablation analysis. We demonstrated the representational capacity of our multi-task learning model by jointly producing keypoints, person bounding boxes and person segmentation results.

CHAPTER 3

SELF-SUPERVISED LEARNING OF 3D HUMAN POSE USING MULTI-VIEW GEOMETRY

3.1 Introduction

Human pose estimation in the wild is a challenging problem in computer vision. Although there are large-scale datasets [18, 19] for two-dimensional (2D) pose estimation, 3D datasets [20, 21] are either limited to laboratory settings or limited in size and diversity. Since collecting 3D human pose annotations in the wild is costly and 3D datasets are limited, researchers have resorted to weakly or self supervised approaches with the goal of obtaining an accurate 3D pose estimator by using minimal amount of additional supervision on top of the existing 2D pose datasets. Various methods have been developed to this end. These methods, in addition to ground-truth 2D poses, require either additional supervision in various forms (such as unpaired 3D ground truth data[22], a small subset of labels [23]) or (extrinsic) camera parameters in multiview settings [24]. To the best of our knowledge, there is only one method [25] which can produce a 3D pose estimator by using only 2D ground-truth. In this paper, we propose another such method.

Our method, "EpiloparPose," uses 2D pose estimation and epipolar geometry to obtain 3D poses, which are subsequently used to train a 3D pose estimator. EpipolarPose works with an arbitrary number of cameras (must be at least 2) and it does not need any 3D supervision or the extrinsic camera parameters, however, it can utilize them if provided. On the Human3.6M [20] and MPI-INF-3DHP [21] datasets, we set the new state-of-the-art in 3D pose estimation for weakly/self-supervised methods.

Human pose estimation allows for subsequent higher level reasoning, *e.g.* in au-

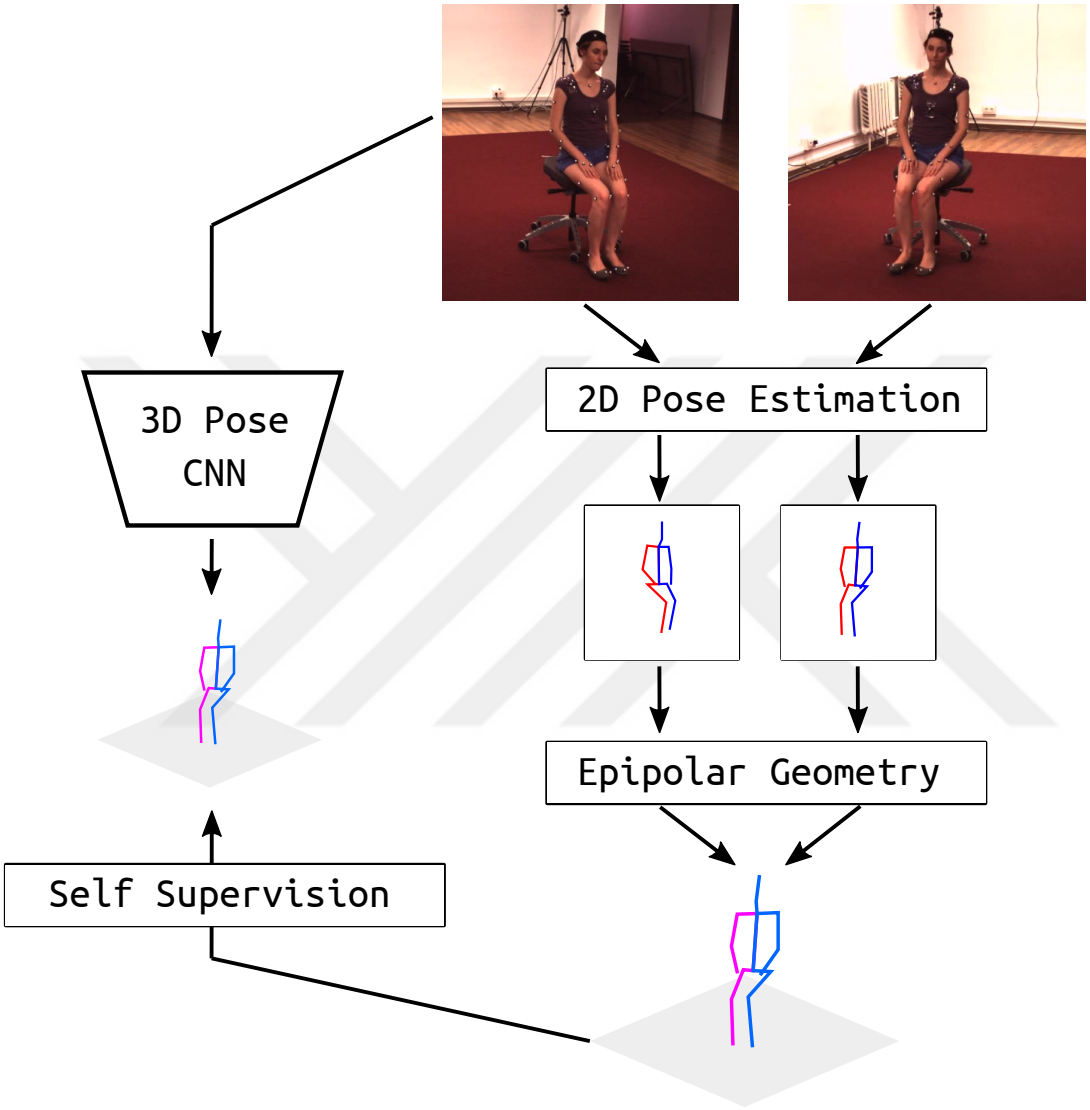

Figure 3.1: **EpipolarPose** uses 2D pose estimation and epipolar geometry to obtain 3D poses which are subsequently used to train a 3D pose estimator.

tonomous systems (cars, industrial robots) and activity recognition. In such tasks, structural errors in pose might be more important than the localization error measured by the traditional evaluation metrics such as MPJPE (mean per joint position error) and PCK (percentage of correct keypoints). These metrics treat each joint independently, hence, fail to asses the whole pose as a structure. Figure 3.4 shows that structurally very different poses yield the same MPJPE with respect to a reference pose. To address this issue, we propose a new performance measure, called the Pose Structure Score (PSS), which is sensitive to structural errors in pose. PSS computes a scale invariant performance score with the capability to score the structural plausibility of a pose with respect to its ground truth. Note that PSS is not a loss function, it is a performance measure that can be used along with MPJPE and PCK to describe the representation capacity of a pose estimator.

To compute PSS, we first need to model the natural distribution of ground-truth poses. To this end, we use an unsupervised clustering method. Let $\mathbf{p}$ be the predicted pose for an image whose ground-truth is $\mathbf{q}$. First, we find which cluster centers are closest to $\mathbf{p}$ and $\mathbf{q}$. If both of them are closest to (*i.e.* assigned to) the same cluster, then the pose structure score (PSS) of $\mathbf{q}$ is said to be 1, otherwise it is 0.

3.1.1 Contributions

Our contributions are as follows:

- We present EpipolarPose, a method that can predict 3D human pose from a single-image. For training, EpipolarPose does not require any 3D supervision nor camera extrinsics. It creates its own 3D supervision by utilizing epipolar geometry and 2D ground-truth poses.

- We set the new state-of-the-art among weakly/self-supervised methods for 3D human pose estimation.

- We present Pose Structure Score (PSS), a new performance measure for 3D human pose estimation to better capture structural errors.

3.2 Related Work

Our method, EpipolarPose, is a single-view method during inference; and a multi-view, self-supervised method during training. Before discussing such methods in the literature, we first briefly review entirely single-view (during both training and inference) and entirely multi-view methods for completeness.

3.2.1 Single-view methods

In many recent work, convolutional neural networks (CNN) are used to estimate the coordinates of the 3D joints directly from images [65, 66, 67, 68, 21]. Li and Chan [69] were the first to show that deep neural networks can achieve a reasonable accuracy in 3D human pose estimation from a single image. They used two deep regression networks and body part detection. Tekin *et al.* [65] show that combining traditional CNNs for supervised learning with auto-encoders for structure learning can yield good results. Contrary to common regression practice, Pavlakos *et al.* [70] were the first to consider 3D human pose estimation as a 3D keypoint localization problem in a voxel space. Recently, "integral pose regression" proposed by Sun *et al.* [71] combined volumetric heat maps with a soft-argmax activation and obtained state-of-the-art results.

Additionally, there are two-stage approaches which decompose the 3D pose inference task into two independent stages: estimating 2D poses, and lifting them into 3D space [72, 73, 74, 75, 76, 72, 67, 21]. Most recent methods in this category use state-of-the-art 2D pose estimators [77, 14, 78, 79] to obtain joint locations in the image plane. Martinez *et al.* [74] use a simple deep neural network that can estimate 3D pose given the estimated 2D pose computed by a state-of-the-art 2D pose estimator. Pavlakos *et al.* [80] proposed the idea of using ordinal depth relations among joints to bypass the need for full 3D supervision.

Methods in this category require either full 3D supervision or extra supervision (*e.g.* ordinal depth) in addition to full 3D supervision.

3.2.2 Multi-view methods

Methods in this category require multi-view input both during and training. Early work [81, 82, 83, 84, 85] used 2D pose estimations obtained from calibrated cameras to produce 3D pose by triangulation or pictorial structures model. More recently, many researchers [86, 87] used deep neural networks to model multi-view input with full 3D supervision.

3.2.3 Weakly/self-supervised methods

Weak and self supervision based methods for human pose estimation have been explored by many [25, 23, 22, 24] due to lack of 3D annotations. Pavlakos *et al.* [24] use a pictorial structures model to obtain a global pose configuration from the keypoint heatmaps of multi-view images. Nevertheless, their method needs full camera calibration and a keypoint detector producing 2D heatmaps.

Rhodin *et al.* [23] utilize multi-view consistency constraints to supervise a network. They need a small amount of 3D ground-truth data to avoid degenerate solutions where poses collapse to a single location. Thus, lack of in-the-wild 3D ground-truth data is a limiting factor for this method [23].

Recently introduced deep inverse graphics networks [88, 89] have been applied to the human pose estimation problem [22, 25]. Tung *et al.* [22] train a generative adversarial network which has a 3D pose generator trained with a reconstruction loss between projections of predicted 3D poses and input 2D joints and a discriminator trained to distinguish predicted 3D pose from a set of ground truth 3D poses. Following this work, Drover *et al.* [25] eliminated the need for 3D ground-truth by modifying the discriminator to recognize plausible 2D projections.

To the best of our knowledge, EpipolarPose and Drover *et al.*'s method are the only ones that do not require any 3D supervision or camera extrinsics. While their method does not utilize image features, EpipolarPose makes use of both image features and epipolar geometry and produces much more accurate results (4.3 mm less error than Drover *et al.*'s method).

3.3 Models and Methods

The overall training pipeline of our proposed method, EpipolarPose, is given in Figure 3.2. The orange-background part shows the inference pipeline. For training of EpipolarPose, the setup is assumed to be as follows. There are n cameras ($n \geq 2$ must hold) which simultaneously take the picture of the person in the scene. The cameras are given id numbers from 1 to n where consecutive cameras are close to each other (*i.e.* they have small baseline). The cameras produce images I_1, I_2, $\dots I_n$. Then, the set of consecutive image pairs, $\{(I_i, I_{i+1}) | i = 1, 2, \dots, n-1\}$, form the training examples.

3.3.1 Training

In the training pipeline of EpipolarPose (Figure 3.2), there are two branches each starting with the same pose estimation network (a ResNet followed by a deconvolution network [71]). These networks were pre-trained on the MPII Human Pose dataset (MPII) [18]. During training, only the pose estimation network in the upper branch is trained; the other one is kept frozen.

EpipolarPose can be trained using more than 2 cameras but for the sake of simplicity, here we will describe the training pipeline for $n = 2$. For $n = 2$, each training example contains only one image pair. Images I_i and I_{i+1} are fed into both the 3D (upper) branch and 2D (lower) branch pose estimation networks to obtain volumetric heatmaps $\hat{H}, H \in \mathbb{R}^{w \times h \times d}$ respectively, where w, h are the spatial size after deconvolution, d is the depth resolution defined as a hyperparameter. After applying soft argmax activation function $\varphi(\cdot)$ we get 3D pose $\hat{V} \in \mathbb{R}^{J \times 3}$ and 2D pose $U \in \mathbb{R}^{J \times 2}$ outputs where J is the number of body joints.

As an output of 2D pose branch, we want to obtain the 3D human pose V in the global coordinate frame. Let the 2D coordinate of the j^{th} joint in the i^{th} image be $U_{i,j} = [x_{i,j}, y_{i,j}]$ and its 3D coordinate be $[X_j, Y_j, Z_j]$, we can describe the relation

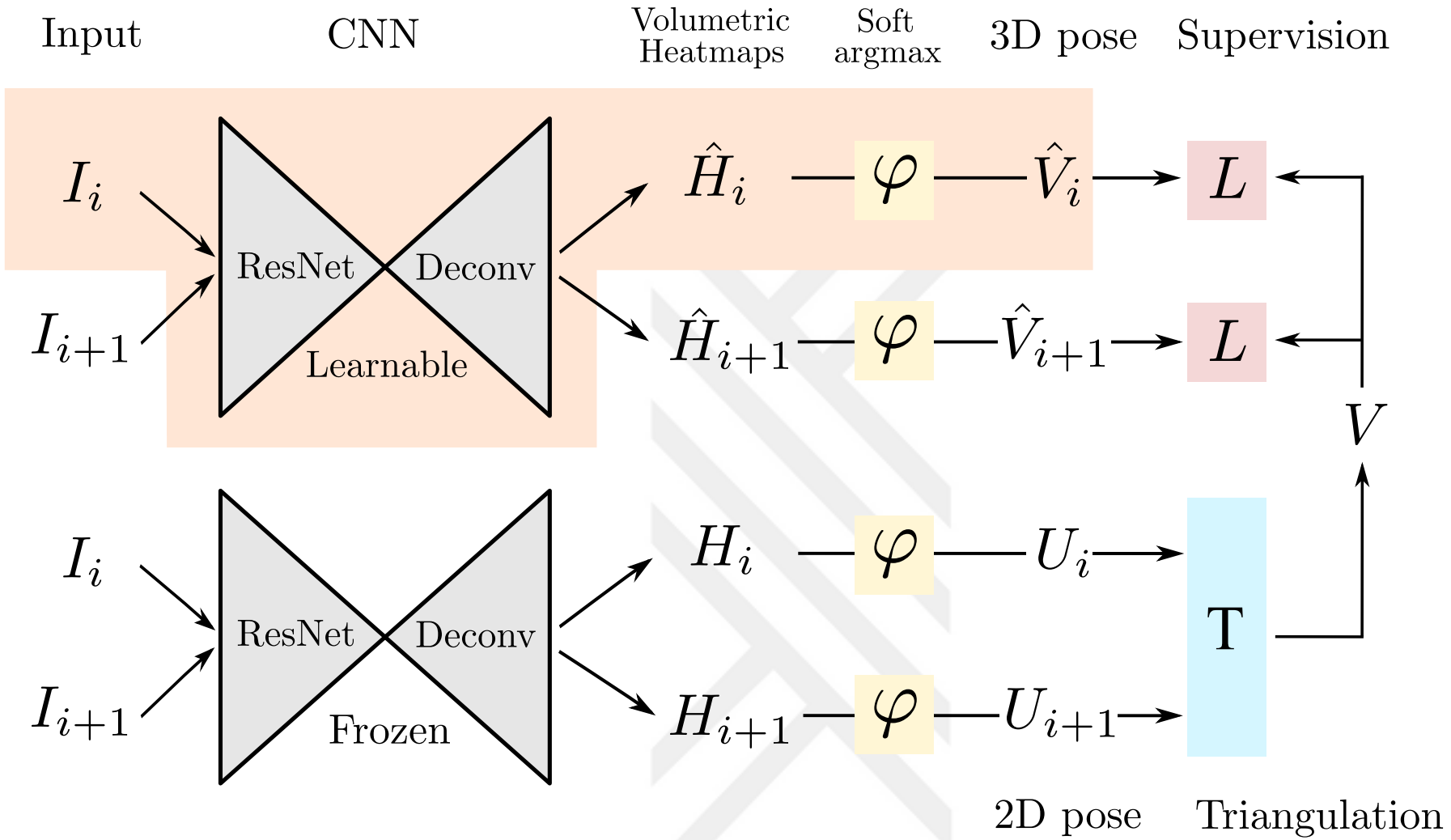

Figure 3.2: **Overall architecture of EpipolarPose during training.** The orange-background part in the upper branch denotes the inference pipeline. During training, EpipolarPose is multi-view: a pair of images (I_i, I_{i+1}) simultaneously taken by two consecutive cameras is fed into the CNN pose estimators. It is also self-supervised: the 3D pose (V) generated by the lower branch using triangulation (*i.e.* epipolar geometry) is used as a training signal for the CNN in the upper branch. During inference (the orange-background part), EpipolarPose is a monocular method: it takes a single image (I_i) as input and estimates the corresponding 3D pose ($\hat{V}_i$). (φ: soft argmax function, T: triangulation, L: smooth L1 loss.)

between them assuming a pinhole image projection model

$$\begin{bmatrix} x_{i,j} \\ y_{i,j} \\ w_{i,j} \end{bmatrix} = K\,[R|RT] \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix}, K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}, \tag{31}$$

where $w_{i,j}$ is the depth of the j^{th} joint in the i^{th} camera's image with respect to the camera reference frame, K encodes the camera intrinsic parameters (*e.g.*, focal length f_x and f_y, principal point c_x and x_y), R and T are camera extrinsic parameters of rotation and translation, respectively. We omit camera distortion for simplicity.

When camera extrinsic parameters are not available, which is usually the case in dynamic capture environments, we can use body joints as calibration targets. We assume the first camera as the center of the coordinate system, which means R of the first camera is identity. For corresponding joints in U_i and U_{i+1}, in the image plane, we find the fundamental matrix F satisfying $U_{i,j}FU_{i+1,j} = 0$ for $\forall j$ using the RANSAC algorithm. From F, we calculate the essential matrix E by $E = K^T FK$. By decomposing E with SVD, we obtain 4 possible solutions to R. We decide the correct one by verifying possible pose hypotheses by doing cheirality check. The cheirality check basically means that the triangulated 3D points should have positive depth [90].

Finally, to obtain a 3D pose V for corresponding synchronized 2D images, we utilize triangulation (*i.e.* epipolar geometry) as follows. For all joints in (I_i, I_{i+1}) that are not occluded in either image, triangulate a 3D point $[X_j, Y_j, Z_j]$ using polynomial triangulation [91]. For settings including more than 2 cameras, we calculate the vector-median to find the median 3D position.

To calculate the loss between 3D pose in camera frame $\hat{V}$ predicted by the upper (3D) branch, we project V onto corresponding camera space, then minimize $\mathrm{smooth}_{L_1}(V - \hat{V})$ to train the 3D branch where

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{32}$$

3.3.1.1 Why do we need a frozen 2D pose estimator?

In the training pipeline of EpipolarPose, there are two branches each of which is starting with a pose estimator. While the estimator in the upper branch is trainable, the other one in the lower branch is frozen. The job of the lower branch estimator is to produce 2D poses. One might question the necessity of the frozen estimator since we could obtain 2D poses from the trainable upper branch as well. When we tried to do so, our method produced degenerate solutions where all keypoints collapse to a single location. In fact, other multi-view methods faced the same problem [23, 92]. Rhodin *et al.* [23] solved this problem by using a small set of ground-truth examples, however, obtaining such ground-truth may not be feasible in most of the in the wild settings. Another solution proposed recently [92] is to minimize angular distance between estimated relative rotation $\hat{R}$ (computed via Procrustes alignment of the two sets of keypoints) and the ground truth R. Nevertheless, it is hard to obtain ground truth R in dynamic capture setups. To overcome these shortcomings, we utilize a frozen 2D pose detector during training time only.

3.3.2 Inference

Inference involves the orange-background part in Figure 3.2. The input is just a single image and the output is the estimated 3D pose $\hat{V}$ obtained by a soft-argmax activation, $\varphi(\cdot)$, on 3D volumetric heatmap $\hat{H}_i$.

3.3.3 Refinement, an optional post-training

In the literature there are several techniques [74, 75, 66] to lift detected 2D keypoints into 3D joints. These methods are capable of learning generalized 2D$\rightarrow$3D mapping which can be obtained from motion capture (MoCap) data by simulating random camera projections. Integrating a refinement unit (RU) to our self supervised model can further improve the pose estimation accuracy. In this way, one can train EpipolarPose on his/her own data which consists of multiple view footages without any labels and integrate it with RU to further improve the results. To make this possible, we modify the input layer of RU to accept noisy 3D detections from EpipolarPose and make it

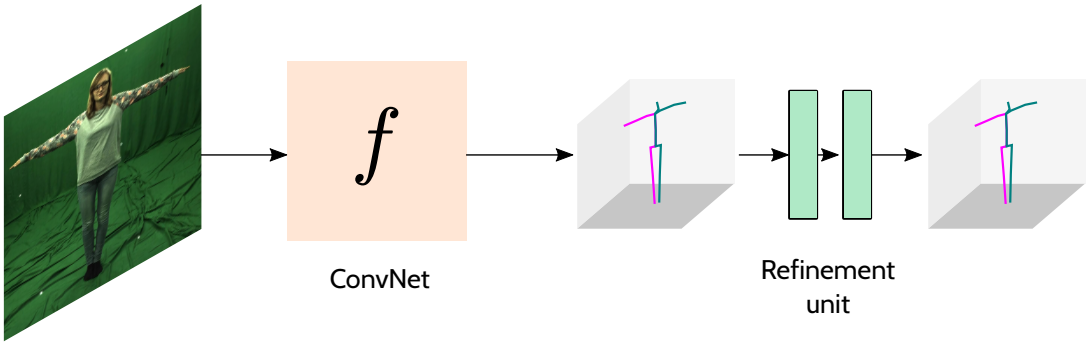

Figure 3.3: Overall inference pipeline with a refinement unit which is an optional stage to refine the predictions of the model trained with self supervision. The f function denotes the inference function (orange-background part in Figure 3.2) of EpipolarPose.

learn a refinement strategy. (See Figure 3.3)

The overall RU architecture is inspired by [74, 75]. It has 2 computation blocks which have certain linear layers followed by Batch Normalization [93], Leaky ReLU [94] activation and Dropout layers to map 3D noisy inputs to more reliable 3D pose predictions. To facilitate information flow between layers, we add residual connections [45] and apply intermediate loss to expedite the intermediate layers' access to supervision.

3.3.4 Pose Structure Score

As we discussed in Section 3.1, traditional evaluation metrics (such as MPJPE, PCK) treat each joint independently, hence, fail to asses the whole pose as a structure. In Figure 3.4, we present example poses that have the same MPJPE but are structurally very different, with respect to a reference pose.

We propose a new performance measure, called the Pose Structure Score (PSS), which is sensitive to structural errors in pose. PSS computes a scale invariant performance score with the capability to assess the structural plausibility of a pose with respect to its ground truth pair. Note that PSS is not a loss function, it is a performance score that can be used along with MPJPE and PCK to describe the representation capacity of a pose estimator. PSS is an indicator about the deviation from the

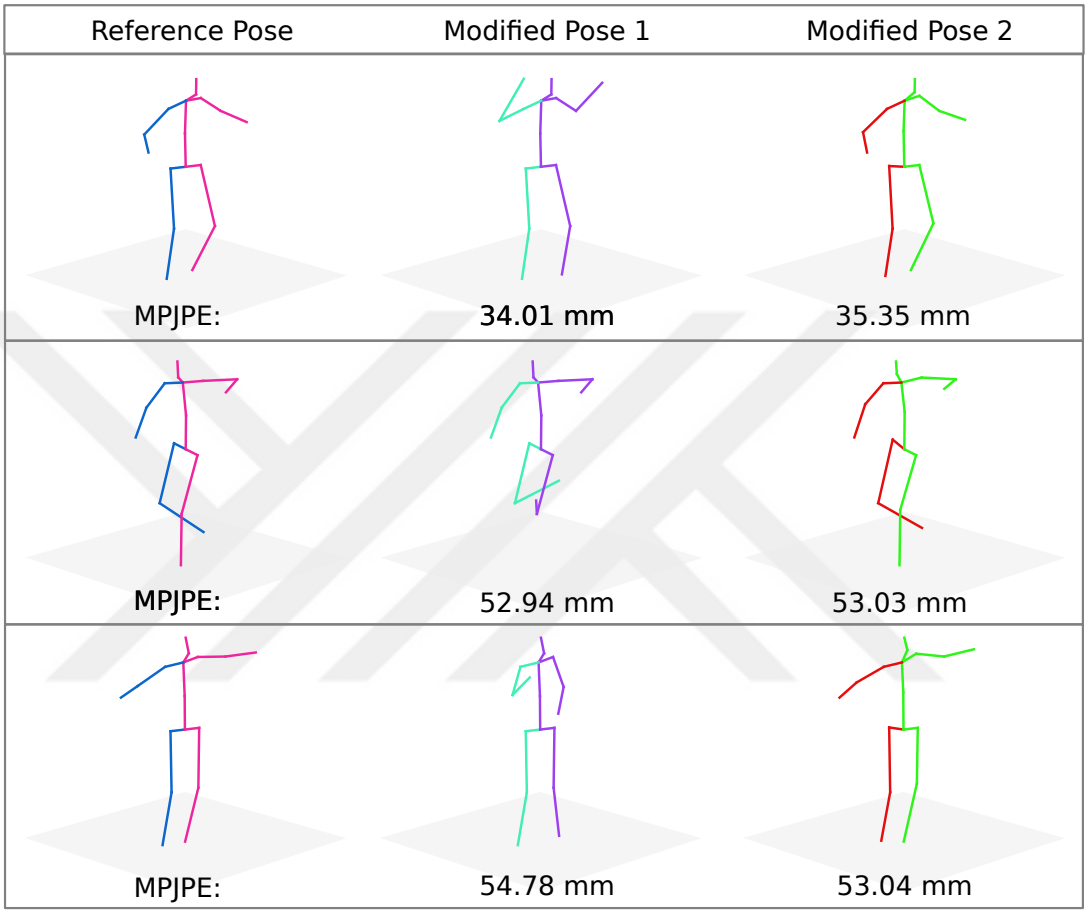

Figure 3.4: **Left**: reference poses from Human3.6M dataset. **Middle**: manually modified poses to obtain similar MPJPE with poses on the *right*, yet structured different from reference poses. **Right**: poses obtained by adding random gaussian noise to each body joints.

ground truth pose that has the potential to cause a wrong inference in a subsequent task requiring semantically meaningful poses, *e.g.* action recognition, human-robot interaction.

3.3.4.1 How to obtain PSS?

Given a ground-truth set composed of n poses $\mathbf{p}_i, i \in \{1, \cdots, n\}$, we normalize each pose vector by $\hat{\mathbf{p}}_i = \frac{\mathbf{p}_i}{||\mathbf{p}_i||}$. Then, we compute k cluster centers $\boldsymbol{\mu}_j, j \in \{1, \cdots, k\}$ using k-means clustering. Then, to compute the PSS of a predicted pose $\mathbf{p}$ against its ground-truth pose $\mathbf{q}$, we use

$$\text{PSS}(\mathbf{p}, \mathbf{q}) = \delta\big(C(\mathbf{p}), C(\mathbf{q})\big) \quad \text{where} \tag{33}$$

$$C(\mathbf{p}) = \arg\min_k ||\mathbf{p} - \boldsymbol{\mu}_k||_2^2, \qquad \delta(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{34}$$

The PSS of a set of poses is the average over their individual scores as computed in Eq. (33). Figure 3.5 shows the t-SNE [95] graph of poses and clusters. Figure 3.6 depicts the cluster centers which represent canonical poses in Human3.6M dataset.

In our experiments, we chose the number of pose clusters as 50 and 100. We denoted the corresponding PSS results with PSS@50 and PSS@100 expressions. Note that PSS computes the percentage of correct poses, therefore higher scores are better.

3.3.5 Implementation details

We use the Integral Pose [71] architecture for both 2D and 3D branches with a ResNet-50 [45] backend. Input image and output heatmap sizes are 256×256 and $J \times 64 \times 64 \times 64$, respectively where J is the number of joints. We initialize all models used in experiments after training on the MPII [18].

During training, we use mini-batches of size 32, each one containing I_i, I_{i+1} image pairs. If more than two cameras are available, we include the views from all cameras

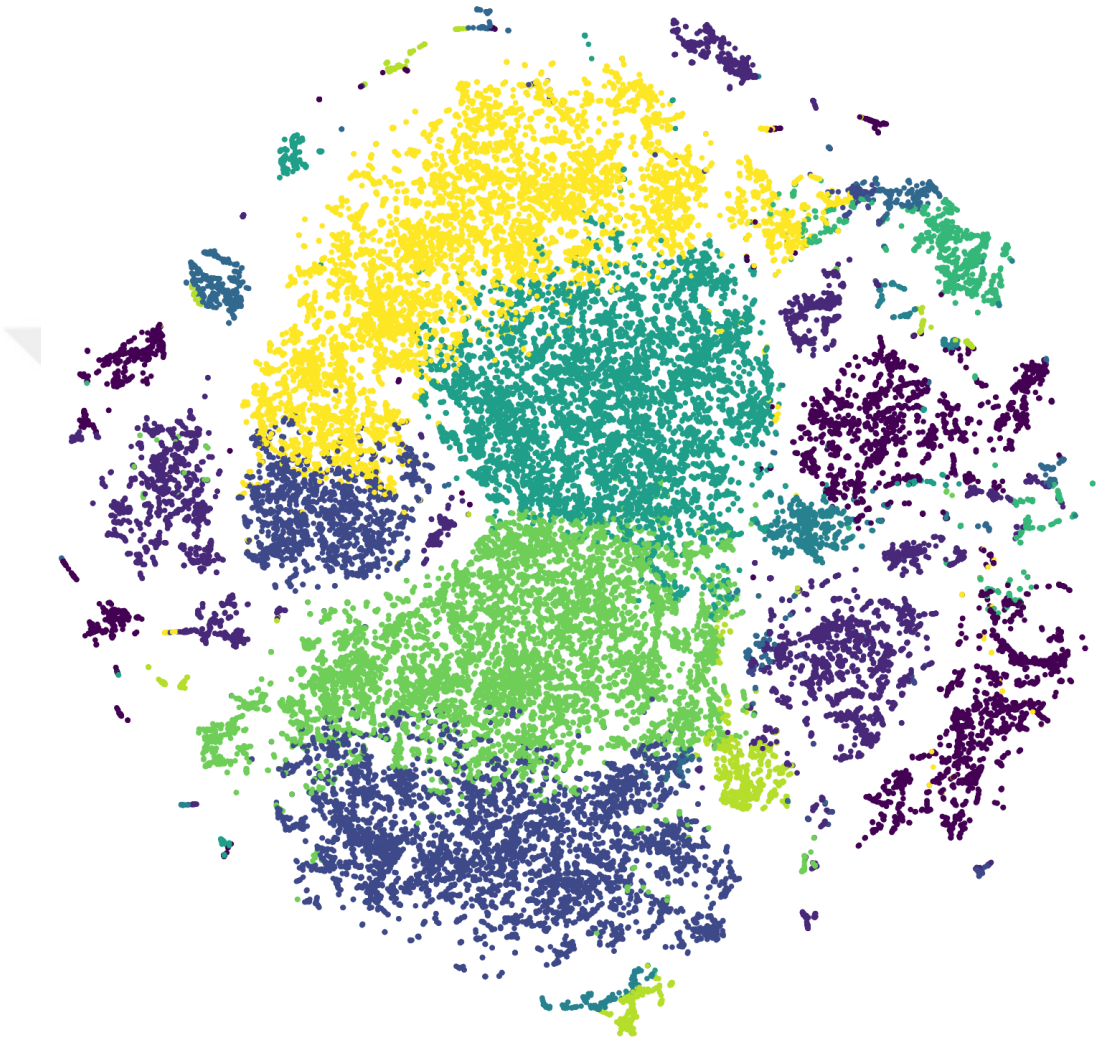

Figure 3.5: **t-SNE graph of human poses after clustering.** Here we choose $k = 10$ for visualization purposes. Each color represents a cluster.

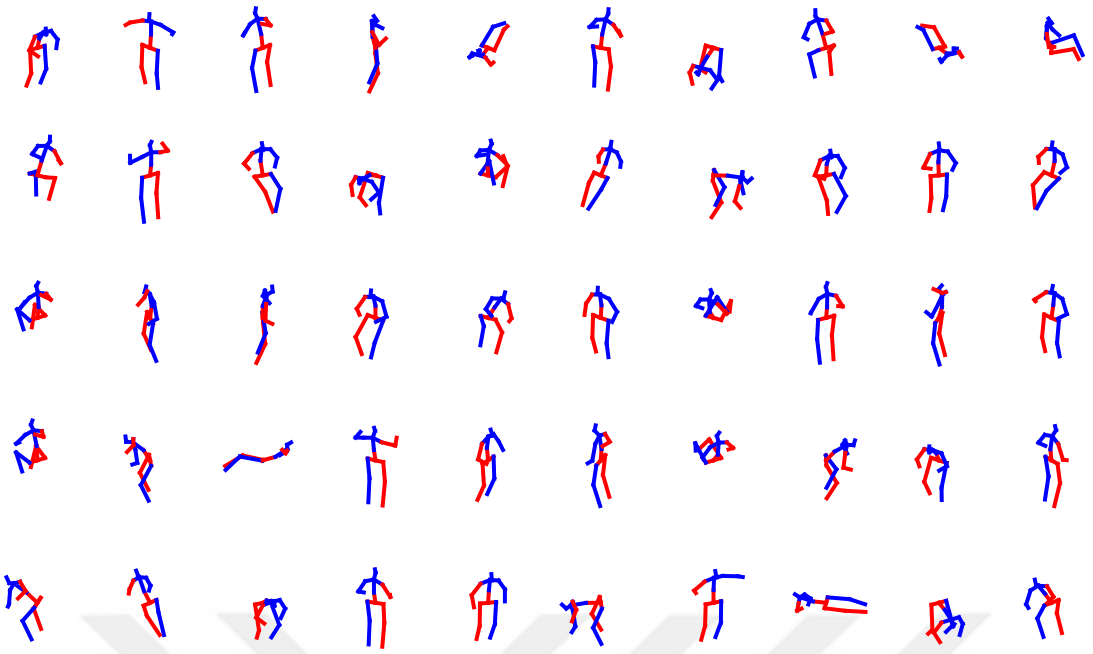

Figure 3.6: **Cluster centers** which represents the canonical poses in Human3.6M ($k = 50$).

in a mini-batch. We train the network for 140 epochs using Adam optimizer [96] with a learning rate of 10^{-3} multiplied with 0.1 at steps 90 and 120. Training data is augmented by random rotations of $\pm 30^{\circ}$ and scaled by a factor between 0.8 and 1.2. Additionally, we utilize synthetic occlusions [97] to make the network robust to occluded joints. For the sake of simplicity, we run the 2D branch once to produce triangulated 3D targets and train the 3D branch using cached labels. We implemented the whole pipeline using PyTorch [98].

3.4 Experiments

3.4.1 Datasets.

We first conduct experiments on the Human3.6M (H36M) large scale 3D human pose estimation benchmark [20]. It is one of the largest datasets for 3D human pose estimation with 3.6 million images featuring 11 actors performing 15 daily activities, such as eating, sitting, walking and taking a photo, from 4 camera views. We mainly use this dataset for both quantitative and qualitative evaluation.

Table 3.1: **Triangulation results on H36M.** Effects of different 2D keypoint sources on triangulation performance. *GT 2D* denotes the usage of ground truth 2D labels. *H36M 2D* and *MPII 2D* shows the pose estimation models trained on those datasets.

Methods	**MPJPE**	**NMPJPE**	**PMPJPE**	**PSS@50**	**PSS@100**
Pavlakos *et al.* [24]	56.89	-	-	-	-
GT 2D	4.38	2.87	2.13	98.93	97.16
GT 2D (w/o $\mathbf{R}$)	n/a	22.46	15.06	98.83	96.03
H36M 2D	28.37	26.28	25.19	95.08	94.2
MPII 2D	45.86	37.79	36.83	90.06	85.96

We follow the standard protocol on H36M and use the subjects 1, 5, 6, 7, 8 for training and the subjects 9, 11 for evaluation. Evaluation is performed on every 64^{th} frame of the test set. We include average errors for each method.

To demonstrate further applicability of our method, we use MPI-INF-3DHP (3DHP) [21] which is a recent dataset that includes both indoor and outdoor scenes. We follow the standard protocol: The five chest-height cameras and the provided 17 joints (compatible with H36M) are used for training. For evaluation, we use the official test set which includes challenging outdoor scenes. We report the results in terms of PCK and NPCK to be consistent with [23]. Note that we do not utilize any kind of background augmentation to boost the performance for outdoor test scenes.

3.4.2 Metrics.

We evaluate pose accuracy in terms of MPJPE (mean per joint position error), PMPJPE (procrustes aligned mean per joint position error), PCK (percentage of correct keypoints), and PSS at scales @50 and @100. To compare our model with [23], we measured the normalized metrics NMPJPE and NPCK, please refer to [23] for further details. Note that PSS, by default, uses normalized poses during evaluation. In the presented results "n/a" means "not applicable" where it's not possible to measure respective metric with provided information, "-" means "not available". For instance,

Table 3.2: **H36M results. Top:** Comparison of results between our methods trained with different settings and the state-of-the-art fully supervised methods. (*FS*: fully supervised, *SS*: self supervised) **Bottom:** Effect of adding refinement unit (RU) over SS. (* uses the 2D keypoints from an MPII pre trained model as input, hence is comparable to our SS+RU model.)

Supervised training on all subjects of H36M

Methods	**MPJPE**	**NMPJPE**	**PMPJPE**	**PSS@50**	**PSS@100**
Nie *et al.* [99] (ICCV'17)	97.5	-	79.5	-	-
Sanzari *et al.* [100] (ECCV'16)	93.1	-	-	-	-
Tome *et al.* [67] (CVPR'17)	88.4	-	-	73.0	58.8
Rogez *et al.* [101] (CVPR'17)	87.7	-	71.6	-	-
Pavlakos *et al.* [70] (CVPR'17)	71.9	-	-	74.05	53.93
Rhodin *et al.* [23] (CVPR'18)	66.8	63.3	51.6	-	-
Martinez *et al.* [74] (ICCV'17)	62.9	-	47.7	78.12	73.26
Pavlakos *et al.* [80] (CVPR'18)	56.2	-	-	80.03	69.18
Sun *et al.* [71] (ECCV'18)	49.6	-	40.6	-	-
Ours FS	52.82	52.49	45.15	84.44	78.67
Ours SS	76.82	75.53	67.48	73.09	64.03
Ours (w/o $\mathbf{R}$)	n/a	78.83	69.54	70.67	62.05

Integrating Refinement Unit with SS trained network on H36M

Methods	**MPJPE**	**NMPJPE**	**PMPJPE**	**PSS@50**	**PSS@100**
Martinez *et al.* [74] (ICCV'2017)*	67.5	-	52.5	-	-
Ours SS + RU	**60.06**	**60.04**	**46.85**	**80.42**	**75.41**

it's not possible to measure MPJPE or PCK when R, the camera rotation matrix, is not available. For some of the previous methods with open source code, we indicate their respective PSS scores. We hope, in the future, PSS will be adapted as an additional performance measure, thus more results will become available for complete comparisons.

3.4.3 Results

3.4.4 Can we rely on the labels from multi view images?

Table 3.1 summarizes triangulation results from different 2D keypoint sources on the H36M dataset. Note that we use training subjects to obtain these results, since our goal is to find out the performance of triangulation on the training data. Overall, the quality of estimated keypoints is crucial to attain better results. If we have the ground truth 2D keypoints and camera geometry, triangulation gives 4.3 mm error and 99% PSS which is near perfect. Lack of camera geometry reduces the PMPJE and PSS@50 by a small amount of 13 mm and 1%, respectively. A pose detector trained on the 2D labels of H36M improves the MPII-pretrained one up to 17 mm and 5%. Note that, it is expected to have slightly worse performance when evaluating the MPII-pretrained detector on the H36M validation set. Data in H36M was captured with markers, and therefore, have high accuracy and consistency in 2D annotations across subject and scenes; on the other hand, the annotations in MPII were done by humans and some of the keypoints are localized differently. For instance, shoulders and hips are closer to edges of the body in the MPII dataset.

Compared to Pavlakos *et al.*'s [24] results, our triangulation using an MPII-pretrained detector is 11mm better in terms of MPJPE.

3.4.5 Comparison to State-of-the-art

In Table 3.2, we present the results of our model with different supervision types in comparison with recent state-of-the-art methods. We present the fully supervised (FS) version of our model to provide a baseline. Our own implementation of "Integral Pose" architecture [71] produced a slightly different result than reported. The difference between our result (52mm) and the reported one [71] (49mm) can be attributed to the authors' 2D-3D mixed training which we refrained from doing in order to decouple 3D pose estimation stage from 2D.

Our self supervised (SS) model performs quite well compared to the recent fully 3D supervised methods [70, 101, 100, 67] which require abundant labeled data to

learn. Obtaining comparable results to state-of-the-art methods without using any 3D ground truth examples is a promising step for such a nontrivial task.

Refinement Unit (RU) which is an optional extension to our SS network is helpful for achieving better results. Adding RU further improves the performance of our SS model by 20% . To measure the representation capacity of the outputs from our SS model, we compare its result with Martinez *et al.*'s work [74]. Since the RU architecture is identical to Martinez *et al.*, we selected their model trained with 2D keypoints from an MPII-pretrained pose detector for a fair comparison. This results show that 3D depth information learned by our SS training method provides helpful cues to improve the performance of 2D-3D lifting approaches.

In Table 3.4 *top*, we show the FS training results on the 3DHP dataset as a baseline. We further use that information to analyze the differences between FS and SS training.

3.4.6 Weakly/Self Supervised Methods

Table 3.3 outlines the performance of weakly/self supervised methods in the literature along with ours on the H36M dataset. The top part includes the methods not requiring paired 3D supervision. Since Tung *et al.* [22] use unpaired 3D ground truth labels that are easier to obtain, we place them here. Our SS model (with or without R) outperforms all previous methods [22, 24] by a large margin in MPJPE metric. We observe a large difference (21mm) between training with ground truth 2D triangulations and MPII-pretrained ones. This gap indicates us that the 2D keypoint estimation quality is crucial for better performance.

To better understand the source of performance gain in ours and Rhodin *et al.*'s, we can analyze the gap between the models trained with full supervision (FS) and subject 1 of H36M and 3DHP only (S1). In our method, the difference between FS and S1 training is 12 and 9mm, while Rhodin *et al.*'s difference is 15 and 18mm for H36M and 3DHP, respectively (lower is better). It shows us that our learning strategy is better at closing the gap. Even though Rhodin *et al.* uses S1 for training, our SS method outperforms it on H36M dataset. In the case of S1 training, there is an explicit improvement (14mm, 4mm for H36M and 3DHP respectively) with our approach. In

addition, SS training with our method on 3DHP has comparable results to Rhodin *et al.*' S1.

Finally, the *bottom* part in Table 3.3 gives a fair comparison of our model against Drover *et al.*'s since they report results only with 14 joints. Our method yields 4mm less error than their approach.

3.5 Conclusion

In this work, we have shown that even without any 3D ground truth data and the knowledge of camera extrinsics, multi view images can be leveraged to obtain self supervision. At the core of our approach, there is EpipolarPose which can utilize 2D poses from multi-view images using epipolar geometry to self-supervise a 3D pose estimator. EpipolarPose achieved state-of-the-art results in Human3.6M and MPI-INF-3D-HP benchmarks among weakly/self-supervised methods. In addition, we discussed the weaknesses of localization based metrics *i.e.* MPJPE and PCK for human pose estimation task and therefore proposed a new performance measure Pose Structure Score (PSS) to score the structural plausibility of a pose with respect to its ground truth.

Table 3.3: **H36M weakly/self supervised results. Top:** Methods that can be trained without 3D ground truth labels. (Tung *et al.* [22] uses unpaired 3D supervision which is easier to get. *3DInterp* denotes the results of [89] implemented by [22]. *2D GT* denotes training with triangulations obtained from ground truth 2D labels.) **Middle:** Methods requiring a small set of ground truth data. (S1 denotes using ground truth labels of H36M subject #1 during training.) **Bottom:** Comparison to Drover *et al.* [25] that evaluated using 14 joints (14j)

Training without ground truth data

Methods	**MPJPE**	**NMPJPE**	**PMPJPE**	**PSS@50**	**PSS@100**
Pavlakos *et al.* [24] (CVPR'2017)	118.41	-	-	-	-
Tung *et al.* - 3DInterp [22] (ICCV'2017)	98.4	-	-	-	-
Tung *et al.* [22] (ICCV'2017)	97.2	-	-	-	-
Ours SS	**76.82**	**75.53**	**67.48**	**73.09**	**64.03**
Ours SS (w/o $\mathbf{R}$)	n/a	78.83	69.54	70.67	62.05
Ours SS (2D GT)	55.69	55.5	48.27	83.9	78.69

Training with only Subject 1 of H36M

Methods	**MPJPE**	**NMPJPE**	**PMPJPE**	**PSS@50**	**PSS@100**
Rhodin *et al.* [23] S1	n/a	78.2	64.6	-	-
Rhodin *et al.* [23] S1 (w/o $\mathbf{R}$)	n/a	80.1	65.1	-	-
Ours S1	**65.58**	**64.92**	**57.22**	**81.91**	**75.2**
Ours S1 (w/o $\mathbf{R}$)	n/a	66.98	60.16	77.65	72.4

Evaluation using 14 joints

Methods	**MPJPE**	**NMPJPE**	**PMPJPE**	**PSS@50**	**PSS@100**
Drover *et al.* [25](14j) (ECCVW'2018)	-	-	64.6	-	-
Ours SS (14j)	**70.09**	**68.15**	**60.27**	n/a	n/a

Table 3.4: **3DHP results. Top:** Fully supervised training results. **Middle:** Self supervised learning using only subject 1. **Bottom:** Self supervised training without any ground truth examples.

Supervised training

Methods	**MPJPE**	**NMPJPE**	**PCK**	**NPCK**	**PSS@50**	**PSS@100**
Mehta *et al.* [21]	-	-	72.5	-	-	-
Rhodin *et al.* [23] FS	n/a	**101.5**	n/a	**78.8**	-	-
Ours FS	**108.99**	106.38	**77.5**	78.1	87.15	82.21

Training with only Subject 1 of 3DHP

Methods	**MPJPE**	**NMPJPE**	**PCK**	**NPCK**	**PSS@50**	**PSS@100**
Rhodin *et al.* [23] S1	n/a	119.8	n/a	73.1	-	-
Rhodin *et al.* [23] S1 (w/o $\mathbf{R}$)	n/a	121.8	n/a	72.7	-	-
Ours S1	n/a	**115.37**	n/a	**74.4**	**75.64**	**73.15**
Ours S1 (w/o $\mathbf{R}$)	n/a	119.86	n/a	73.5	73.41	70.97

Training without ground truth data

Methods	**MPJPE**	**NMPJPE**	**PCK**	**NPCK**	**PSS@50**	**PSS@100**
Ours SS	126.79	125.65	64.7	71.9	70.94	67.58

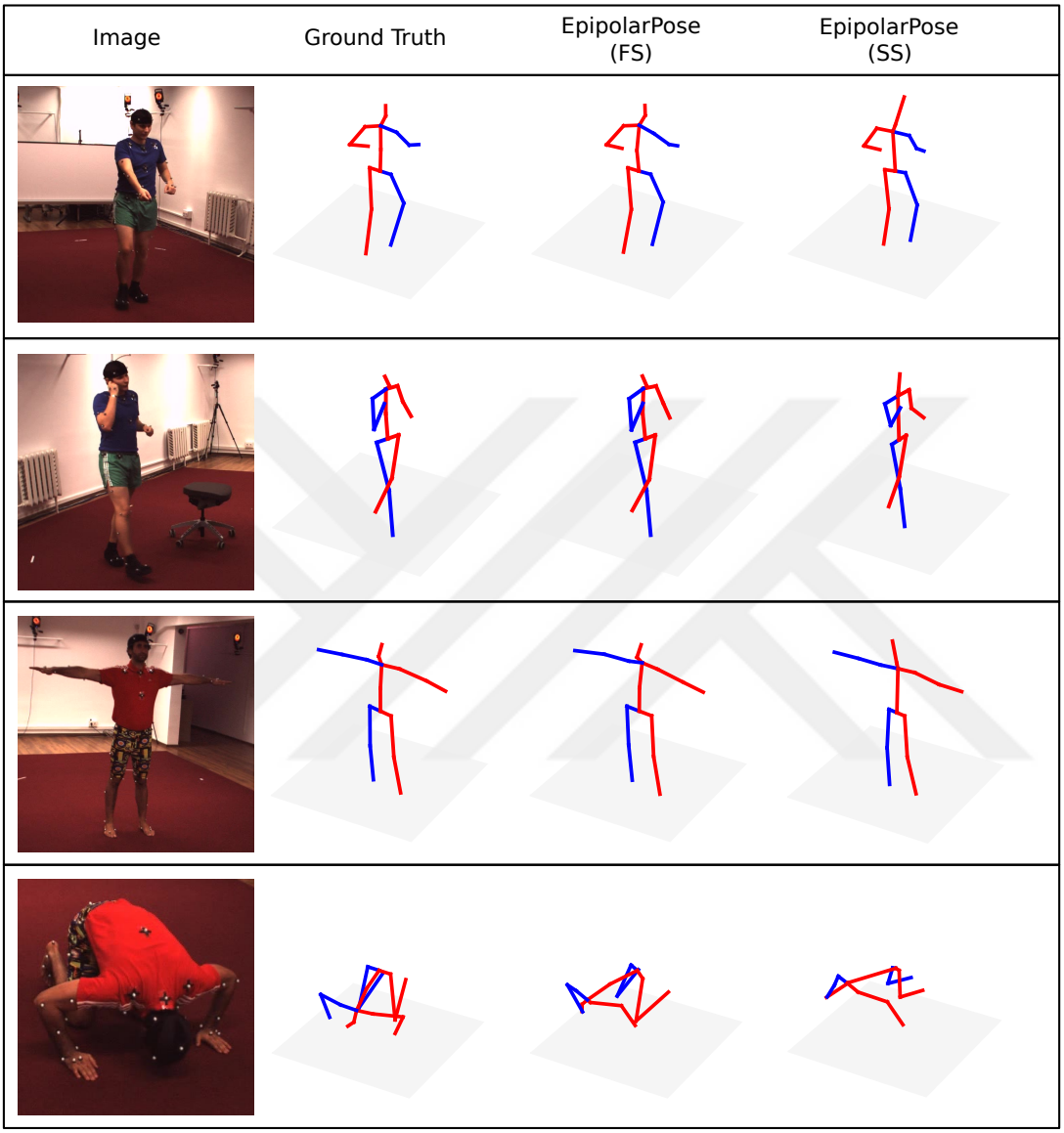

Figure 3.7: **Qualitative results on H36M dataset.** Provided 3D poses are from different camera views for better visualization. Last row depicts a failure case. (*FS:* fully supervised training, *SS:* self supervised training)

CHAPTER 4

CONCLUSIONS

In chapter 2, we introduced the Pose Residual Network that is able to accurately assign keypoints to person detections outputted by a multi task learning architecture (MultiPoseNet). Our pose estimation method achieved state-of-the-art performance among bottom-up methods and comparable results with top-down methods. Our method has the fastest inference time compared to previous methods. We showed the assignment performance of pose residual network ablation analysis. We demonstrated the representational capacity of our multi-task learning model by jointly producing keypoints, person bounding boxes and person segmentation results.

In chapter 3, we have shown that even without any 3D ground truth data and the knowledge of camera extrinsics, multi view images can be leveraged to obtain self supervision. At the core of our approach, there is EpipolarPose which can utilize 2D poses from multi-view images using epipolar geometry to self-supervise a 3D pose estimator. EpipolarPose achieved state-of-the-art results in Human3.6M and MPI-INF-3D-HP benchmarks among weakly/self-supervised methods. In addition, we discussed the weaknesses of localization based metrics *i.e.* MPJPE and PCK for human pose estimation task and therefore proposed a new performance measure Pose Structure Score (PSS) to score the structural plausibility of a pose with respect to its ground truth.

REFERENCES

[1] M. R. Ronchi and P. Perona, "Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation," *International Conference on Computer Vision*, 2017.

[2] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *European Conference on Computer Vision*, 2014.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] 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," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," *European Conference on Computer Vision*, 2016.

[6] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," *European Conference on Computer Vision*, 2016.

[7] U. Iqbal and J. Gall, "Multi-person pose estimation with local joint-to-person associations," *European Conference on Computer Vision Workshops*, 2016.

[8] G. Ning, Z. Zhang, and Z. He, "Knowledge-Guided Deep Fractal Neural Networks for Human Pose Estimation," *IEEE Transactions on Multimedia*, 2017.

[9] A. Newell, Z. Huang, and J. Deng, "Associative Embedding: End-to-End Learning for Joint Detection and Grouping," *Advances in Neural Information Processing*, 2017.

[10] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded Pyramid Network for Multi-Person Pose Estimation," *arXiv preprint arXiv:1711.07319*, 2017.

[11] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards Accurate Multi-person Pose Estimation in the Wild," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *International Conference on Computer Vision*, 2017.

[13] H. Fang, S. Xie, Y. Tai, and C. Lu, "RMPE: Regional Multi-Person Pose Estimation," *International Conference on Computer Vision*, 2017.

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

[15] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," *European Conference on Computer Vision*, 2016.

[16] C.-J. Chou, J.-T. Chien, and H.-T. Chen, "Self Adversarial Training for Human Pose Estimation," *arXiv preprint arXiv:1707.02439*, 2017.

[17] S. Huang, M. Gong, and D. Tao, "A Coarse-Fine Network for Keypoint Localization," *International Conference on Computer Vision*, 2017.

[18] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *European Conference on Computer Vision*, 2014.

[20] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2014.

[21] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3D human pose estimation in the wild using improve cnn supervision," *International Conference on 3DVision*, 2017.

[22] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, "Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision," *International Conference on Computer Vision*, 2017.

[23] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua, "Learning monocular 3d human pose estimation from multi-view images," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[24] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting multiple views for marker-less 3d human pose annotations," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] D. Drover, R. MV, C.-H. Chen, A. Agrawal, A. Tyagi, and C. P. Huynh, "Can 3d pose be learned from 2d projections alone?," *European Conference on Computer Vision Workshops*, 2018.

[26] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[27] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[28] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2013.

[29] S. Johnson and M. Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation," *British Machine Vision Conference*, 2010.

[30] M. Andriluka, S. Roth, and B. Schiele, "Pictorial Structures Revisited: People Detection and Articulated Pose Estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[31] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human Pose Estimation Using Body Parts Dependent Joint Regressors," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[32] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "Using k-poselets for detecting people and localizing their keypoints," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[33] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[34] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation," *Advances in Neural Information Processing*, 2014.

[35] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human Pose Estimation with Iterative Error Feedback," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[36] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-Context Attention for Human Pose Estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[37] I. Lifshitz, E. Fetaya, and S. Ullman, "Human Pose Estimation using Deep Consensus Voting," *European Conference on Computer Vision*, 2016.

[38] V. Belagiannis and A. Zisserman, "Recurrent Human Pose Estimation," *International Conference on Automatic Face and Gesture Recognition*, 2017.

[39] V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines," *European Conference on Computer Vision*, 2014.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[41] L. Ladicky, P. H. Torr, and A. Zisserman, "Human Pose Estimation Using a Joint Pixel-wise and Part-wise Formulation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[42] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik, "Articulated pose estimation using discriminative armlet classifiers," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[43] S. Varadarajan, P. Datta, and O. Tickoo, "A Greedy Part Assignment Algorithm for Realtime Multi-Person 2D Pose Estimation," *arXiv preprint arXiv:1708.09182*, 2017.

[44] U. Iqbal, A. Milan, and J. Gall, "PoseTrack: Joint Multi-Person Pose Estimation and Tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[46] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2017.

[47] F. Xia, P. Wang, A. Yuille, and L. Angeles, "Joint Multi-Person Pose Estimation and Semantic Part Segmentation in a Single Image," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[48] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[49] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *International Conference on Computer Vision*, 2017.

[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *European Conference on Computer Vision*, 2016.

[52] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[53] R. Girshick, "Fast R-CNN," *International Conference on Computer Vision*, 2015.

[54] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing*, 2015.

[55] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.

[56] F. Chollet *et al.*, "Keras," 2015.

[57] H. Gaiser, M. de Vries, A. Williamson, Y. Henon, M. Morariu, V. Lacatusu, E. Liscio, W. Fang, M. Clark, M. V. Sande, and M. Kocabas, "fizyr/keras-retinanet 0.2," 2018.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[60] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," *International Conference on Machine Learning*, 2009.

[61] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, 2015.

[62] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[63] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," *arXiv preprint arXiv:1802.02611*, 2018.

[64] A. Kendall, V. Badrinarayanan, , and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *British Machine Vision Conference*, 2017.

[65] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, "Structured prediction of 3D human pose with deep neural networks," *British Machine Vision Conference*, 2016.

[66] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2D and 3D image cues for monocular body pose estimation," *International Conference on Computer Vision*, 2017.

[67] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3D pose estimation from a single image," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[68] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," *International Conference on Computer Vision*, 2017.

[69] S. Li and A. B. Chan, "3D human pose estimation from monocular images with deep convolutional neural network," *Asian Conference on Computer Vision*, 2014.

[70] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[71] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," *European Conference on Computer Vision*, 2018.

[72] C.-H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[73] F. Moreno-Noguer, "3D human pose estimation from a single image via distance matrix regression," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[74] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," *International Conference on Computer Vision*, 2017.

[75] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3D pose estimation," *Association for the Advancement of Artificial Intelligence*, 2018.

[76] X. Zhou, M. Zhu, K. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3D human pose estimation from monocular video," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[77] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[78] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *European Conference on Computer Vision*, 2016.

[79] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," *European Conference on Computer Vision*, 2018.

[80] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3D human pose estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[81] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele, "Multi-view pictorial structures for 3d human pose estimation," *British Machine Vision Conference*, 2013.

[82] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnorr, "A study of parts-based object class detection using complete graphs," *International Journal of Computer Vision*, 2010.

[83] M. Burenius, J. Sullivan, and S. Carlsson, "3D pictorial structures for multiple view articulated pose estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[84] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D pictorial structures for multiple human pose estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[85] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D pictorial structures revisited: Multiple human pose estimation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2016.

[86] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[87] A. Elhayek, E. de Aguiar, A. Jain, J. Thompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "MARCOnI—ConvNet-based MARker-less motion capture in outdoor and indoor scenes," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2017.

[88] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," *Advances in Neural Information Processing*, 2015.

[89] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, "Single image 3d interpreter network," *European Conference on Computer Vision (ECCV)*, 2016.

[90] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2004.

[91] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, 1997.

[92] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi, "Discovery of latent 3d keypoints via end-to-end geometric reasoning," *Advances in Neural Information Processing*, 2018.

[93] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Journal of Machine Learning Research*, 2015.

[94] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *International Conference on Machine Learning*, 2013.

[95] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, 2008.

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[97] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe, "How robust is 3d human pose estimation to occlusion?," *IROS Workshop - Robotic Co-workers 4.0*, 2018.

[98] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," *International Conference on Learning Representations*, 2017.

[99] B. Xiaohan Nie, P. Wei, and S.-C. Zhu, "Monocular 3d human pose estimation by predicting depth on joints," *International Conference on Computer Vision*, 2017.

[100] M. Sanzari, V. Ntouskos, and F. Pirri, "Bayesian image based 3d pose estimation.," *European Conference on Computer Vision*, 2016.

[101] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net: Localization-classification-regression for human pose," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.