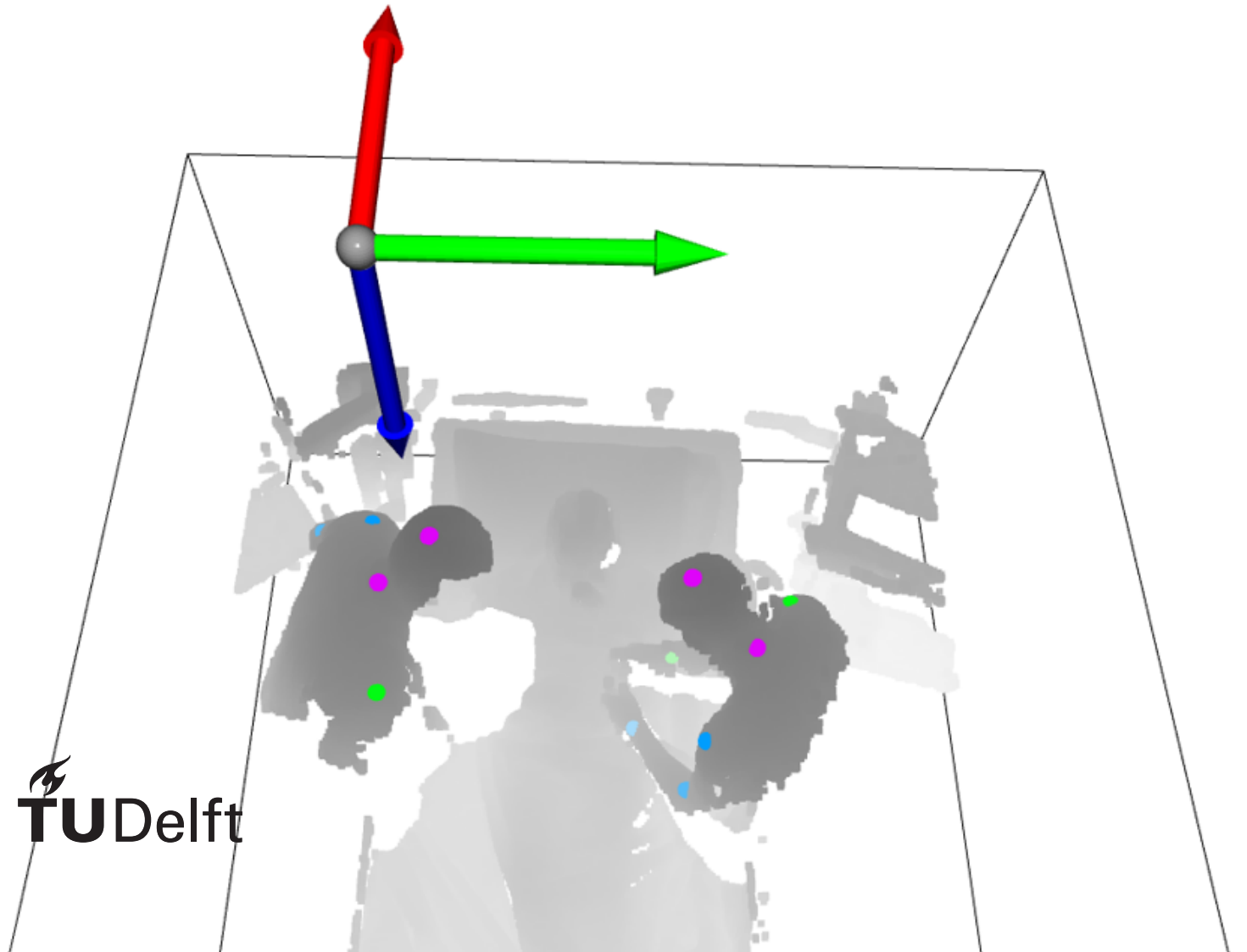
3D Human Pose Estimation

Using a Top-View Depth Camera

Prerak P Mody

MSc Thesis
Computer Science
May 19th, 2020



TU Delft

3D Human Pose Estimation

Using a Top-View Depth Camera

by

Prerak P Mody

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended privately on Tuesday May 19, 2020 at 03:00 PM.

Student number:	4777042	
Project duration:	September 1, 2019 – May 29, 2020	
Thesis committee:	Prof.Dr. E. Eisemann,	TU Delft, Professor, Chair
	Dr. K. Hildenbrandt,	TU Delft, Supervisor
	Dr. H. Hung,	TU Delft, Associate Professor
	Dr. E. van der Heide,	Philips Research, Supervisor
	Dr. F. Zuo,	Philips Research, Supervisor

This thesis is confidential and cannot be made public until May 19, 2022.

TU Delft

Preface

The motivation for this research originated from my desire to evaluate how one can apply machine learning algorithms outside of carefully curated academic datasets and in real-world situations. This thesis project, done at Philips Research, Eindhoven, has given me such an opportunity by testing the efficacy of deep learning algorithms in indoor hospital scenarios. This work has taught me how one can test recent technologies on existing problems and take a scientific approach to such experimentation.

For this, I would like to express my gratitude and respect to my industrial supervisors, Dr. Esther van der Heide and Dr. Fei Zuo from Philips Research. Their guidance, patience and constant endeavor to challenge me have improved my research and engineering skills. I would also like to thank my academic supervisor, Dr. Klaus Hildebrandt for his constant support and motivation through the duration of this work. A special mention goes out Dr. Erik Bresch from Philips Research whose technical inputs have been useful at multiple stages of this project and to Kevin Chang, a fellow intern, also supervised by Dr. Esther and Dr. Fei.

Finally, I would like to thank my partner, Sailee Sansgiri for her presence and motivational support and to my parents for having supported my masters study.

Prerak P Mody
Delft, May 2020

Abstract

The onset of delirium, a disturbance in the mental activities of a patient, can be potentially detected by understanding activities within an Intensive Care Unit (ICU) room. Such activities can be extracted by estimating human pose via a visual capture of the scene. This work uses a top-view depth camera in an ICU room to estimate pose of the non-patient stakeholders. The top-view leads to self-occlusions of body joints and thus poses a challenge for estimation of complete human pose. In addition, the presence of multiple persons in the room poses a secondary challenge, as detected body-joints need to be parsed into individual poses. To address these challenges, a 3D point cloud is extracted from the top-view depth image and passed through a 3D Convolutional Neural Network (CNN). This baseline method is capable of estimating both body-joints and body-parts to eventually output human pose for multiple persons. To improve the quality of output poses, the baseline method can benefit from additional spatial context since the problem of human pose estimation has a highly structured output. The proposed techniques either increase the receptive field, perform feature extraction at multiple scales or change the order of data processing. An increase in F1-score for the proposed methods highlights the importance of additional spatial context as a crucial tool to improve the performance of pose estimation models.

Contents

1 Introduction

Delirium is a mental condition that can affect the pace of recovery for individuals under medical care[50], [1]. Detection of the early onset of this complication is a problem that has so far been unsolved [48]. Understanding the context around a patient in an ICU room might help provide insight into the patients medical status. Such a context can be understood by studying the activities and interactions between the various clinical stakeholders. The extraction of different kinds of activities has the potential to be automated, due to the wide availability of cheap visual sensing hardware along with the recent success of machine learning technologies. One method to simplify the complexity of a medical scene is to analyze the human pose of various stakeholders. Here, human pose refers to the localization of anatomical joints of the human body and thus, can also be seen as a low-dimensional representation of humans in a visual capture of a scene. It can be used as an additional input for action recognition or person-tracking algorithms to help with the understanding of context within an ICU room.

Since monitoring of individuals in an ICU room involves privacy concerns, it is important to use a visual sensing technique such as a depth camera that is capable of masking identity. Thus, data from an existing experimental setup in multiple ICU rooms of a hospital is used to investigate the extraction of human pose using only a top-view depth camera placed on the ceiling. The output of the depth camera is used as input to an algorithm that estimates the articulated 3D human pose of the non-patient stakeholders. The top-view of the camera causes self-occlusion of body-joints which leads to the depth images containing only partial visual information on the human body. The scenarios in this dataset also contain multiple non-patient stakeholders performing a variety of clinical activities in rooms with variable background clutter. Thus, the collected dataset is much more complex than existing depth camera datasets due to its top-view and the presence of multiple persons.

The estimation of 3D human pose comes with various challenges due to clothing, size, shape, environmental factors and most importantly the high degrees of freedom offered to all body joints. Inter and intra-occlusions, unusual poses and visual sensor noise provide additional real-world challenges to estimate 3D human pose. These challenges lead to a requirement of a 3D human pose estimation model that is able to encapsulate all the above variables in a general framework.

Thus, the first research question of this work is to identify a baseline model and investigate whether depth data collected from a top-view camera can be used for robust estimation of human pose for multiple persons. This baseline model should be able to disambiguate human bodies from the variable background clutter of each ICU room and estimate their human pose.

Furthermore, human pose estimation is a structured problem with strong inter-dependencies between body-joints as defined by a kinematic chain. For example, a model should be able to understand the symmetry of the human body and be able to discriminate between its left and right sides. Thus, another research question of this work is to investigate whether spatially contextual cues can improve the performance of the baseline model such that it understands the the position of one body-joint with respect to another.

This report contains the following sections – Chapter 2 covers the background and related work on human pose estimation. Chapter 3 details the collected dataset, theoretical aspects of the baseline and proposed model with their results discussed in Chapter 4. Finally, Chapter 5 ends the report with some discussions and conclusions.

2

Background and Related work

This chapter details a history of human pose estimation in both two and three dimensions to establish an understanding of the various approaches. The concept of human pose is explained in section 2.1 followed by the methods built and evolved by the research community in section 2.2. This report classifies the approaches into two categories - classical methods shown in subsection 2.2.1 and deep learning methods shown in subsection 2.2.2. Section 2.3 discusses the depth and RGB datasets that have been built to push the boundaries of pose estimation in different real-world scenarios. Finally, various evaluation metrics are explained in section 2.4.

2.1. Human Pose

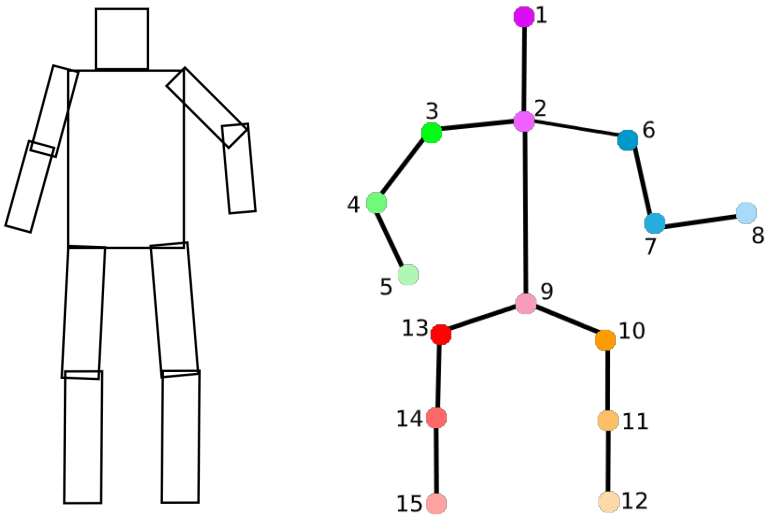
Human pose estimation has applications in fields such as human-computer interaction, security and healthcare. Early applications of human pose estimation were driven by its need in character animation for movie productions. In human computer interaction, pose can be used to directly infer the humans gesture such as in gaming consoles or to extract the subtle cues from their pose (i.e. human-robotic interaction). It can also be used to track the movement of people in sensitive areas for security purposes. Recently, its also found use in sports for analysing the movements of athletes to ensure they dont strain their muscles. Finally, human pose estimation can also be applied in healthcare processes such as monitoring in surgery rooms to optimize worflows or for automated evaluation of physical therapy.

The use of human pose started with the use of calibrated marker-based motion capture systems, where multiple tracker tags were attached to various parts of a human body. These tags are then tracked using multiple cameras to gain accurate human pose. Such a setup, which is time-consuming, complicated and expensive, prohibits applications across a wide range of use cases. Thus, there is a strong requirement for human pose estimation methods which use only raw visual inputs as they are markerless in nature.

A model for human pose can be either using primitive shapes such as rectangles and cylinders or by a simple stick-figure as shown in figure 2.1. This report mainly refers to the stick-figure model where body joints are connected via body parts. The human pose has a fixed kinematic chain which refers to how the various body joints are connected to each other in a tree-like structure. This kinematic chain defines the succession of different body parts which also have constraints on the angles between them. One can also notice that the human body also possesses a left-right symmetry.

The task of *articulated* human pose estimation via visual input(s) involves two steps - 1) finding the location of various body joints and 2) parsing these body joints to infer the limbs (or parts) of the body. This is a challenging task due to the high degrees of freedom available to a human body coupled with other factors such as variations in human clothing along with the clutter and lighting of the environment. These problems can be compounded when there are multiple persons in a scene that can lead to both inter-person and intra-person occlusions.

It should be noted that human pose estimation is usually not the final goal and can be used as an input for other tasks such as activity recognition or person-tracking since it can serve as an intermediate and low-dimensional representation of a human bodies in a scene.



(a) (b)

Figure 2.1: Different models of human pose are shown as either a a) primitive shape model of the human body with the various shapes connected in a defined kinematic chain or b) a stick figure model of the human body representing the various anatomical joints of interest.

2.2. Methods

The task of human pose estimation has been performed using either model-based, discriminative or hybrid methods. Model-based methods use template priors of the human pose while discriminative methods rely only on learning visual cues from datasets. Within each of the these methods, the task has been either to extract 2D or 3D human pose and at times both.

2.2.1. Classical methods

Work on human pose estimation was started by relying on model-based methods that made certain assumptions on the human body which were then embedded into an algorithmic framework. Early methods modeled both image features of the various body parts as well as the spatial dependencies between them in an probabilistic framework.

2.2.1.1. RGB-based approaches

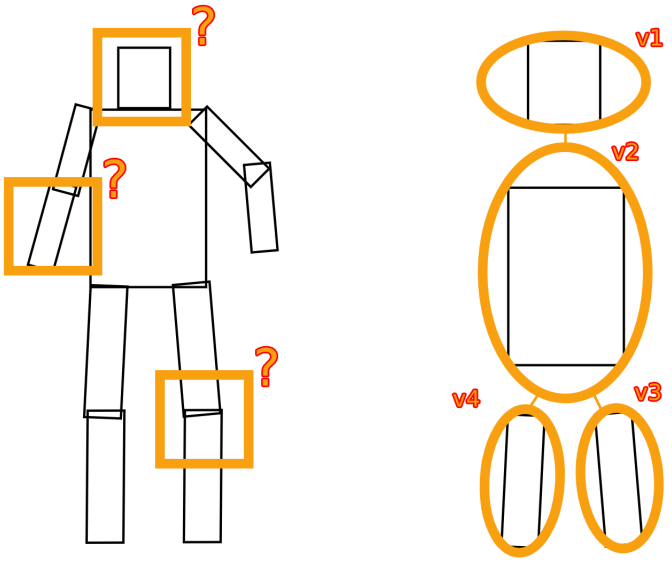
RGB-based pose estimation approaches primarily relied on the Pictorial Structures Model [18] and its various variants [90].

Pictorial Structure Model

Pictorial structures model (PSM) [18] is a minimization technique applied to pose estimation which was developed in the early days of computer vision. This was then widely applied to various works due to the its reformulation as probabilistic technique [15]. The original pictorial structures model defines an object (i.e. *human*) as made of rigid parts (i.e. *limbs*) which have geometric constraints between them (example - angle between head and shoulders). This model attempts to solve the following minimization problem, which is a combination of an *appearance model* and a *configuration model*:-

$$L^* = argmin_L\left[\sum m_i(l_i) + \sum d_{ij}(l_i, l_j)\right] \tag{2.1}$$

Here L^* is the final pose configuration, m_i is the matching cost, d_{ij} is the deformation cost and the limbs are modelled as $l_i = (x_i, y_i, s_i, \theta_i)$. Here x_i, y_i denote the spatial location, s_i denotes the body part scale and θ_i indicate the body part orientation. The matching cost within the *appearance model* captures local information while the deformation cost within the *configuration model* captures global context by modeling spatial relationships between body parts. One can capture local information using image features which can help identify the different body parts. Thus, the optimal pose would be one that minimizes the matching cost for all body parts as well as the deformation cost between two body parts as shown in figure 2.2.



(a) Matching Cost (b) Deformation Cost

Figure 2.2: The building blocks of Pictorial Structure Model (PSM) - matching cost and deformation cost shown on a human pose model. In a) the model tries to find the optimal location of the face to reduce matching cost. In b) the model attempts to globally adjust the detected body parts (modelled as vertices of a graph) to reduce deformation cost.

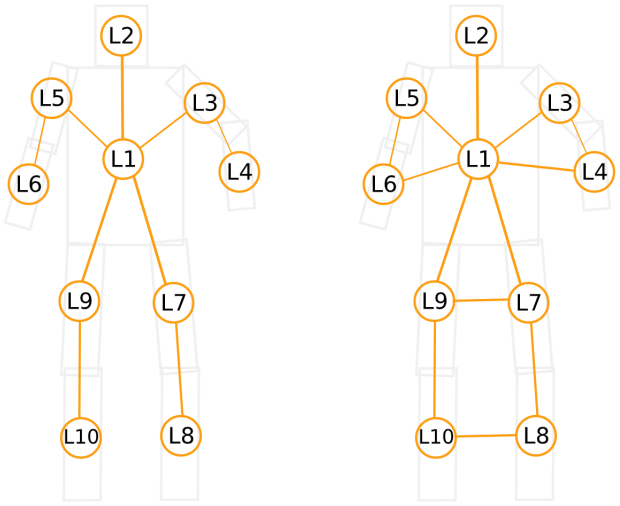
Since such a minimization problem would most likely involve the need for a good initialization of L, [15] instead came up with a simpler probabilistic version of the same which is as follows:-

$$
\begin{aligned}
p(L|I) &= p(I|L).P(L) \\
p(L|I,\theta) &\propto p(I|L,\theta).p(L,\theta) \\
p(L|I,\theta) &\propto p(I|L,u).p(L,c) \\
p(L|I,\theta) &\propto \prod_{i=1..n} p(I|l_i,u_i). \prod_{(v_i,v_j \in E)} p(l_i,l_j|c_{ij})
\end{aligned}
$$

Here L is the pose configuration, I is the image, $\theta = (u,c)$ are model parameters, l_i is a body part, E is the kinematic graph of the human body with v_i, v_j as the two vertices with an edge (or body part between them) and l_i is a limb which starts with v_i as the parent body joint. This formulation now captures the appearance model ($p(I|l_i,u_i)$) as a unary term and the configuration model ($p(l_i,l_j|c_{ij})$) as a pairwise term in a probabilistic framework. Notice that the appearance model has its constituents as the product of all local body part templates. Such a model is trained on a given image dataset via maximum likelihood estimation (to estimate the parameters - u, v) and inference for a new image is performed via maximum aposteriori estimation of $p(L|I,\theta)$.

The earliest approaches used background subtraction and static image features [15],[17] for the appearance models and geometric primitives to model the body for the configuration model in either tree-based or non-tree-based graphical models as seen in figure 2.3. The tree-based approaches [15], [2] [3] modeled the spatial relationships between the adjacent body parts using a kinenematic chain as shown in figure 2.1. Non-tree or graph-based methods [34],[73], [76], [67],[57] embellished the tree models with extra edges to model occlusions, symmetry and long-range relationships. Further improvements were made by either utilizing better image features in the appearance model or advanced body priors in the configuration model. One way to extract better image features is to use discriminatively trained body part detectors such as SVMs [36], random forests [12] or AdaBoost [4]. Other works have used advanced body part priors [83], [90] or proposed efficient probabilistic inference algorithms [72], [66].

Thus, the above approach has used a graphical model to capture local and discriminative body joint information as well as global contextual information to constrain the distributions of these body joints.



(a) Tree Model (b) Non-Tree Model

Figure 2.3: Modeling the spatial relationships between body parts in the deformation cost as a)tree structure or b)non-tree structure.

2.2.1.2. Depth based approaches

Time-of-flight (ToF) cameras can help build a privacy-preserving system that is also capable of providing rich visual cues. Depth sensors within such a visual sensing system provide depth measurements at every pixel in the field of view of the camera. It is especially useful to discriminate the salient parts of a scene from the background clutter. Early work that used depth-images make assumptions on the model of the person in the scene as well as the persons location [24]. Various works impose temporal constraints to solve minimization problems which involve reducing the distance between the depth image (or extracted 3D point cloud) and an assumed human pose model [89], [85]. Other works [21], [25] use the Iterative Closest Point (ICP) algorithm for the same minimization, though they suffer from initialization issues.

One of the earliest successes in pose estimation using depth images without a body model assumption was by [69]. To aid with the task of pose estimation, they performed semantic segmentation (i.e. pixel-wise classification) on depth image pixels to detect masks for body parts. The mean-shift algorithm is used to find a local mode of the detected body part masks to output the location of a body joint. They designed a tree regression method that could extract human pose on a frame-by-frame basis and hence did not suffer from initialization issues. This was done by generating a large corpus of synthetic depth images with a variety of poses for humans with different body characteristics. Their discriminative approach to pose estimation translated into real time processing on commodity GPUs. Similar work using regression trees can be found in Random Tree Walks [87], where a regression tree is trained for each body joint which provides some candidate directions to the next joint in the human kinematic chain.

Other methods [30] combine a tree-based method with graphs as defined by the pictorial structures model to localize body joints. One of their contributions was to propose a novel shape descriptor for 3D point clouds which is different from depth-based [69] or geodesic-distance based features [60].

Thus, classical depth-based approaches have experimented with both 2D and 3D approaches as well as approaches with and without the standard pictorial structures model.

2.2.2. Deep Learning methods

Recently deep learning methods have performed exceptionally well [82] across a range of computer vision tasks like image classification [41], object detection [22], semantic segmentation [28] as well as human pose estimation [8], [80]. The success of deep neural nets (DNN) or specifically convolutional neural nets (CNN) has led to their use as the primary component in human pose estimation methods over the last five years. This technique has shown breakthrough performance on various 2D [5], [46], [33] and 3D human pose estimation [32] benchmarks without the use of any explicitly modeled human body priors. While classical approaches to pose estimation [2] design and optimize individual components (i.e. individual body part detectors), neural networks are capable of end-to-end learning [1] and hence can potentially perform better than classical approaches.

2.2.2.1. Convolutional Neural Networks

A Neural network (f) is a modeling framework that attempts to understand the relationship between an input (x) and output (y). It is loosely modelled around the neurons in a mammalian brain where neurons are connected to each other in a complex network. The connections between these neurons allows for hierarchical features to be learned from the input, especially when layers of neurons are stacked one after the other. Using other tools such as non-linear activation functions allows for non-linear relationships to be understood between the input (x) and output (y). The weights between individual neurons can be altered based on some objective function by using the backpropagation algorithm (a form of gradient-based-learning) [63].

A special type of neural network called convolutional neural network (CNN) is primarily used for computer vision tasks such as image classification, object detection, semantic segmentation and keypoint detection. Their architecture enables them to process array-based inputs such as 2D images or 3D volumes. These networks have two main operations - convolutions and pooling which typically occur one after the other in a series of layers. A convolutional layer is usually initialized with some random image filters and as the network is trained via backpropagation, these filters modify their internal values to learn image features such as edges. On the other hand, a pooling layer reduces the spatial spread of the features or activation maps created by the learned convolutional filters to make the knowledge learned by the filters as translation invariant, an important requirement for visual understanding. Another advantage of using convolutional mechanisms instead of a traditionally fully connected network is that it also reduces the parameters to be learned, thus making training potentially simpler.

2D CNN

When one performs a set of 2D convolutions on an input image, the 2D convolutional kernel is moved in two dimensions along the image. Element-wise multiplication and addition outputs values to form the next layer of image features (i.e. activation maps) as seen in figure 2.4. Thus, if we have C convolutional kernels being applied on an image, we are creating C different interpretations of the input image.

3D CNN

A simple extension of the 2D CNN is the 3D CNN, where a 3D convolutional kernel is moved in 3 dimensions along a 3D input grid.

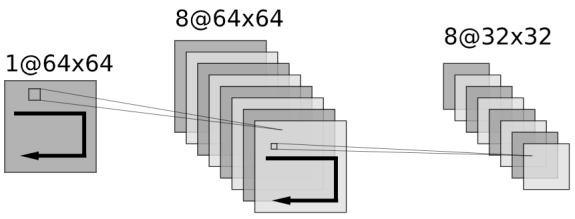


Figure 2.4: The figure shows a sample convolutional and pooling sequence. Eight convolutional kernels of spatial dimensions - 3 x 3 being applied on an image (with 1 pixel padding) to output eight activation maps of the same resolution. A pooling layer with kernel size of 2x2 then reduces the spatial dimensions of the activation maps by half.

[1] An end-to-end learning system is one whose internal parameters are differentiable and hence can be updated via gradient descent

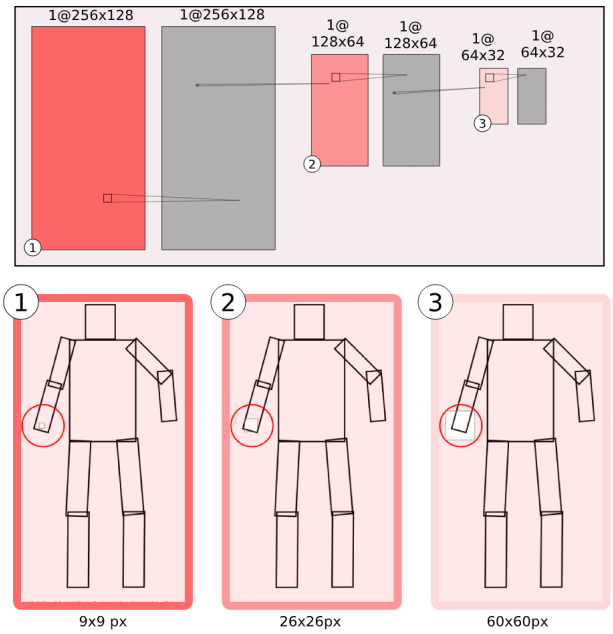


Figure 2.5: The figure shows the increasing receptive field (*in pixels*) of subsequent 9 x 9 convolutional kernels as one goes deeper in a network. Different shades of red represent activation maps created by convolutional operations while the gray activations maps are created due to pooling layers. In Step 1, the 9x9 kernel has a receptive field of the same size while in the subsequent layer (after convolutional and pooling), it has a larger receptive field of 26 x 26.

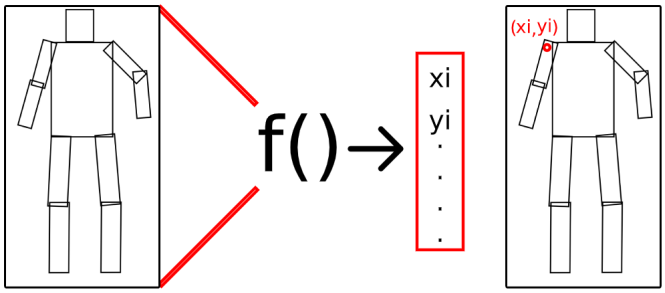


Figure 2.6: The figure shows a convolutional neural network *f()* (consisting of a cascade of convolutional, pooling and fully connected layers) that maps an image $I \in \mathbb{R}^{HxW}$ to 2D pose coordinates $X \in \mathbb{R}^{2J}$. Such a coordinate regression based methodology also performs a cascade of iterative improvements on the original predictions [75].

2.2.2.2. RGB-based CNN's

RGB-based pose estimation CNN's can be understood by studying them through the properties of either their output format or their input processing strategies. The output format can either be a 2D/3D coordinate or a heatmap grid through which 2D/3D coordinates are extracted. For input processing strategies, a model could either detect a person via a bounding box detection approach and then perform pose estimation, or directly detect body-joint keypoints and parse them into a human pose.

Coordinate vs Heatmap Regression

The seminal work by [75] showcased how a deep hierarchy of features in a CNN could be used to model 2D human pose estimation as a regression problem. They show how one can use a CNN model in an iterative refinement framework to obtain more precise estimates of the 2D coordinates of anatomical joints of interest. Unlike graphical methods which used local image features for each body-joint, their CNN used the entire image to establish context for the task of localization of these body-joints. A CNN is able to do so in spite of convolutional kernels being only a fraction of the image as kernels deep in the network have a very large receptive field as seen in figure 2.5.

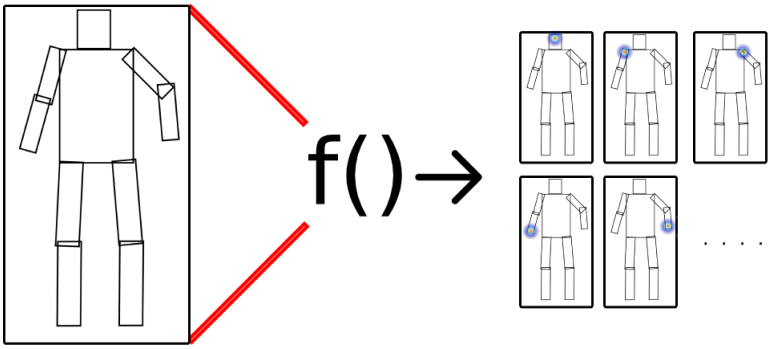


Figure 2.7: A representative network flow diagram that shows the different heatmaps predicted for various body joints by taking as input an image and passing it through a neural network (*f()*). It is shown graphically that the input image resolution is larger than the output heatmap resolution.

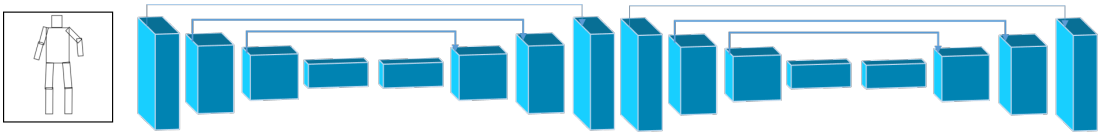


Figure 2.8: A representative architecture of the Hourglass [55] network that combines information from multiple stages in a multi-iterative manner.

But the direct regressing of 3D joints has two drawbacks – high non-linearity [2] and requirement to know in prior the number of people in the input image. To abate this problem, [74] instead propose to use heatmap regression instead of direct coordinate regression. The heatmaps serve as the target for each body-joint where it is modeled as a gaussian centered around the body-joint with a small variance. Thus, instead of predicting an exact location, the network predicts a confidence area as it attempts to localize body joints as seen in figure 2.7.

Using the above three design components – 1) neural networks as feature extractors, 2) iterative refinement and 3) heatmap outputs for individual body joints, others have proposed more complex building blocks for deeper neural architectures. The work by [55] observed that in previous works most of the errors emerged due to joints being assigned to positions quite far away from their ground truth, and hence refinement at a local level shall not be needed to improved model performance. They design an hourglass module which is able to capture and combine information across multiple resolutions in the hope to capture more spatial context on a scene. Added refinement to output is done by stacking multiple hourglass modules together that use the output of a previous hourglass module as shown in figure 2.8.

Top-Down vs Bottom-Up Approaches

In a multi-person pose estimation context, various architectures have been successful at extracting human pose, though they do so in a top-down manner i.e. estimation of bounding box followed by estimation of human pose [28], [80]. This has some disadvantages if the predicted bounding boxes are not accurate enough and hence such a procedure may truncate parts of the body. The runtime of such an approach is also proportional to the number of people in an image which is an undesirable trait. An alternate approach is to detect all body joints together and then parse them into persons [58], [59], [7], [8]. Although CNN-based joint estimation is quite fast, it is also important to ensure that joint-parsing has comparable runtimes. To that end [7], [40] proposes to also learn the 2D vector fields of the body parts between the joints which allows for a greedy joint-parsing for pose formation and hence efficient runtimes.

Other techniques

Another technique of pose refinement is done by predicting the error made in the previous iteration instead of the joint locations [9]. This technique follows the iterative procedure principle used in many works of pose estimation but with the goal of learning the dependencies between the input and output spaces. Given the

[2] a typical image would have a dimension around $\mathbb{R}^{256x256}$ which needs to be mapped to $\mathbb{R}^2$ or $\mathbb{R}^3$

structured nature of human pose estimation, this method can motivate the CNN to learn feature hierarchies that encapsulate properties of the human kinematic chain. Others have also explored a multi-task learning approach where human pose estimation is just one of the many visual tasks the neural network needs to learn. [23] learns the human pose combined with person detection and action recognition with the aim of speeding up the processing times (i.e. training and inferencing) for N tasks by N-times. [43] instead assumes the bounding box of a person and performs multi-task learning by finding a bounding box around the body joint and also outputting its 3D coordinates. [62] builds a multi-task network by performing both 2D and 3D human pose estimation on the input images. They estimate candidate bounding box regions for humans and attach a set of fixed anchor poses to these regions. These anchor poses are then refined as per the individuals present in the bounding box regions.

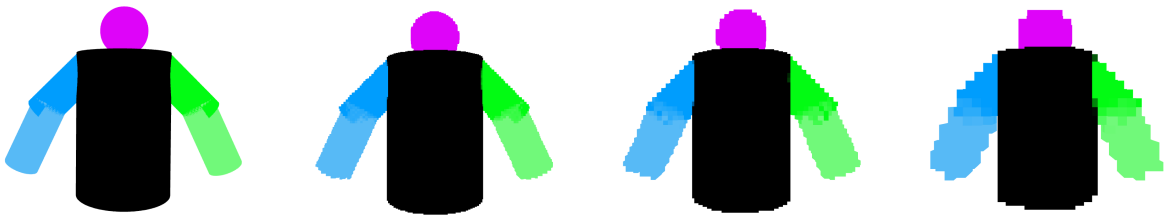
Thus, neural nets have managed to make incremental strides in pose estimation results without the use of the classical pictorial structures prior. Also, since the low-level operations of a neural net have as of yet been difficult to interpret, incorporating explicit body priors such as geometric constraints has not yet been possible.

2.2.2.3. Depth-based CNN's

Depth-image based neural network methods have taken two primary approaches to extract discriminative features – 1) using a 2D depth-image or 2) using a 3D point cloud extracted from the depth image.

2D Input

One of the first works to deploy a neural network in testing their efficacy on depth images was the methodology released with the ITOP dataset [27]. The authors developed a network that would be able to learn discriminative features of a depth image scene regardless of the view from which it was captured. Their network extracts 3D point clouds of the different body joints to obtain viewpoint-invariant high dimensional features and a combination of these local patches and iterative inferencing outputs the final pose. This method is very similar to the pictorial structures model explained in section 2.2.1 which also models individual body joints as local information. To model global pose information they perform iterative refinement of the predicted pose using a recurrent neural network. This work models each body-joint separately and not in a global context as can be done with CNN-based approaches. Hence it may not be suitable to robustly extract human pose in complex scenes. Another approach is to use pose exemplars from a dataset and perform a weighted combination of these poses to yield the pose of the input image [49]. The reasoning to adopt such an approach is that although the theoretical space of human pose is quite large, the space of real-world poses is much smaller. A downside of this approach is to have exemplars which are suitable for the poses in the captured scenes and may not generalize to unseen poses. Methods using 2D depth maps have also been inspired from RGB-based methods such as in [52] which simply adopt the OpenPose method proposed in [7]. The difference exists in their feature extractors which are shallow compared to RBG-based networks since depth images do not contain as much texture and color information as RGB images. Thus, a network with lesser parameter count is also capable of extracting scene information. This is a particularly useful insight on the processing of depth images using CNNs.



(a) Original Point Cloud (resolution=1mm)
(b) Voxelized Point Cloud (resolution=10cm)
(c) Voxelized Point Cloud (resolution=20cm)
(d) Voxelized Point Cloud (resolution=40cm)

Figure 2.9: Voxelization for different voxel resolutions done via the binary occupancy grid method. Increasing the voxel resolution makes the body surfaces more coarse or *block-like* in nature.

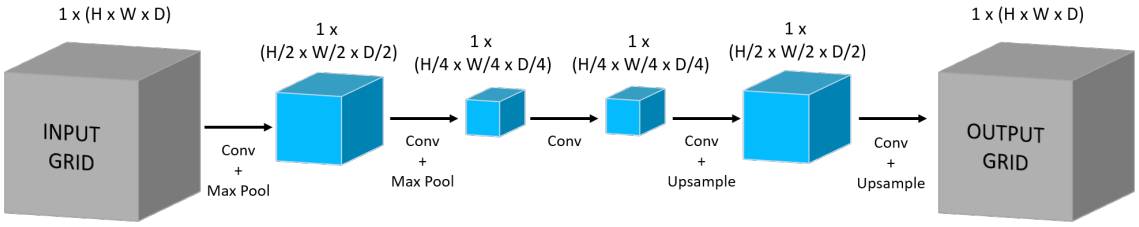


Figure 2.10: A representative neural architecture of the method used in [53]. Here, the input grid is the voxelized point cloud and the output grid are the predicted body-joint heatmaps. The network downsamples the input to a latent representation and then upsamples it back to its original resolution.

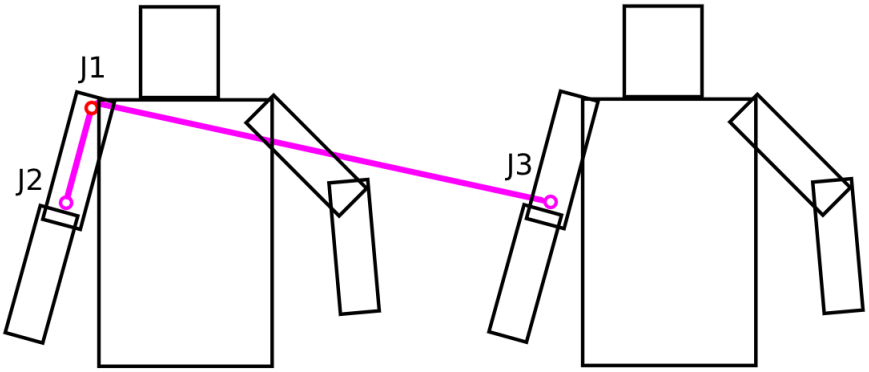


Figure 2.11: The problem of body-joint parsing to extract body-parts is represented in this figure. Specifically, the parent joint J1 is to be connected to either the child joint J2 or J3.

3D Input

The first CNN that took as input 3D point clouds extracted from a depth image [53], voxelized [3] the point clouds as seen in figure 2.9 since 3D CNNs can only process lattice-like inputs. The network then estimates the probability of a joint for each voxel in the output space and a non-maximal suppression extracts the 3D location of a joint. Their hypothesis for choosing a 3D input was that processing 2D depth images for 3D human pose forces the chosen model to also learn the inherent perspective distortion in 2D. This method is not suitable for multi-person pose estimation and has only been tested on a single person pose dataset. It uses an encoder-decoder architecture to output only body-joint heatmaps as shown in figure 2.10. This method cannot be extended to a multi-person scenario as it does not offer any information on how one can parse the predicted body joints into body parts (figure 2.11) to output the human pose of each individual in the scene. A similar approach was taken in [77] which implements a 3D version of the bottom-up pose estimation approach taken in [7]. This method, as of the writing of this report, is the top-performing method on the ITOP dataset [27] as seen in table 2.1. Its network architecture is also suitable for multi-person pose estimation and was hence chosen as a baseline for this report.

Method	Input Representation	Building Block	mean AP
Shotton, 2011 [69]	2D	Random Forests	0.474
Jung, 2015 [87]	2D	Random Forests	0.682
Haque, 2016 [27]	2D	2D CNN + RNN	0.755
Xiong, 2019 [81]	2D	2D CNN	0.805
Moon, 2018 [53]	3D	3D CNN	0.834
Vasileiadis, 2019 [77]	3D	3D CNN	0.845

Table 2.1: Depth-based methods along with their input representation, building block and mean average precision (mAP) on the ITOP dataset

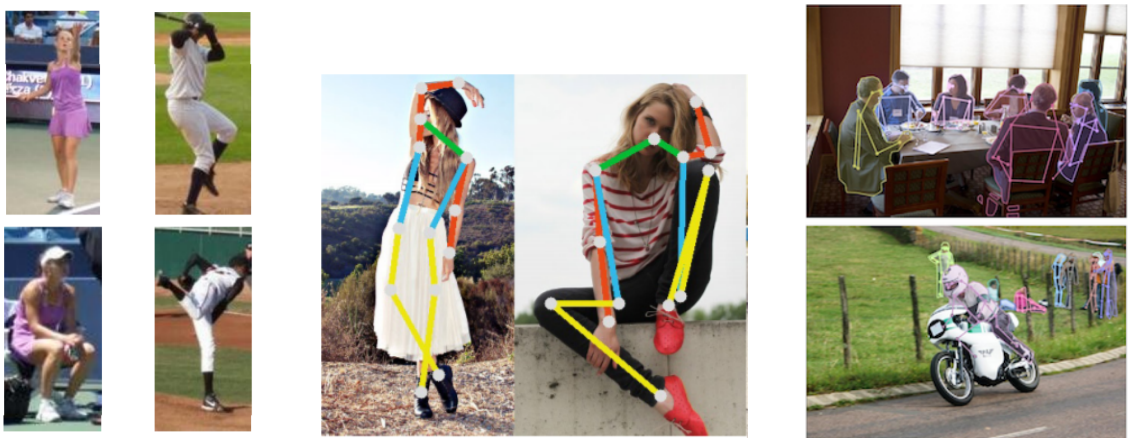
[3] Voxelization is usually done by fixing a voxel size and creating a binary occupancy grid

2.3. Human Pose Datasets

Since computer vision tasks are benchmarked using datasets, following the development of datasets over time provides a good insight into the research trajectory for the task of interest. The surveyed datasets offer different levels of complexity for human pose estimation based on factors such as environmental conditions (lighting and clutter), camera viewpoint, person count, inter and intra-person occlusions, count of annotated body joints and type of activity. The sections below list some of the more commonly used datasets which contain either color images (i.e. RGB) or depth-images.

2.3.1. RGB datasets

RGB (Red-Green-Blue) or color datasets capture the scenarios using easily available commercial camera systems. The creators of these datasets refer to online corpuses of such images and collate and annotated them to create challenging real-world pose estimation datasets.



(a) Leeds Sport Persons (LSP) dataset [35]

(b) FashionPose dataset [35]

(c) MS-COCO dataset [46]

Figure 2.12: Representative images from 2D Human Pose datasets with single and multi-person scenes shown using RGB images

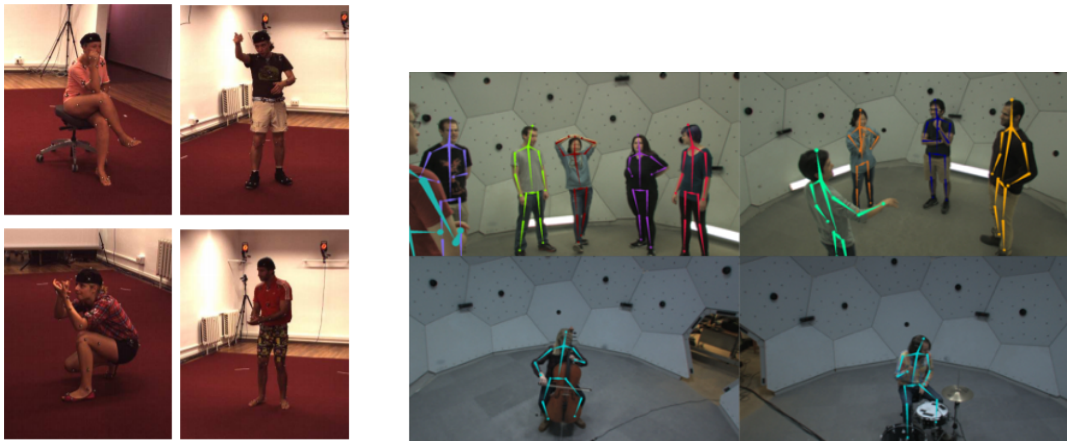
2D Pose Datasets

2D human pose datasets either contain specific scenarios like TV Shows (e.g. Buffy [16]), cinema (e.g. FLIC [65]), sports (e.g. LSP [35]) and fashion (e.g. FashionPose [11]) or more general scenarios like in the Parse [61], MPII [5], PASCAL VOC [14], MS-COCO [46] and Posetrack [33] datasets. Datasets such as Posetrack [33] offer annotations for both pose estimation and tracking. While some datasets offer full-body pose annotations, others offer only top-body annotations [16], [65]. Figure 2.12 shows some sample images from RGB datasets that focus on either single person or multi-person pose estimation.

3D Pose Datasets

Since annotation of body joints in 3D is not a trivial task, one option is to capture them in indoor scenarios by multiple cameras and motion capture hardware. The collection of such datasets started with HumanEva-I [70] which was inspired by its goal to halt the practice of qualitative evaluation of pose estimation methods in the early 2000's. The captured poses using this technique are choreographed with little variations in clothing and a lack of occlusions, hence making them not completely natural. A similar but larger dataset, Human3.6M [32] was released to produce systems that can be used in the real world by providing a corpus of 3.6 million annotated images. It captures a set of fixed poses and expands the dataset by projecting those poses on synthetic graphics models placed in various mixed reality scenarios. Since attaching motion capture markers on the bodies of subjects might make them aware of their pose, the CMU Panoptic dataset [37] eliminates this factor by capturing social interactions within a constructed dome dotted with inward-facing cameras. Due to the availability of high number of view points, the dataset is able to generate high fidelilty point clouds of the entire scene as well. Figure 2.13 shows the indoor setups of the Human3.6M (with person markers) and CMU Panoptic datasets (with a highly dense camera setup).

There also exist other multi-view datasets such as the Shelf and Campus [6] datasets, but these are hand-annotated and hence not large in size. In an effort to capture 3D Pose in outdoor scenes, the KTH Multiview Football [38] dataset uses 3 cameras in a football stadium to extract poses of individual players. Another in-the-wild 3D Pose dataset is 3DPW [79], which acquired 3D pose by combining a video feed and IMU's worn by participants, but the acquired annotations are not of the highest possible quality.



(a) Human 3.6M dataset [32]

(b) CMU Panoptic dataset [37]

Figure 2.13: Representative images from 3D Human Pose datasets captured in indoor scenarios

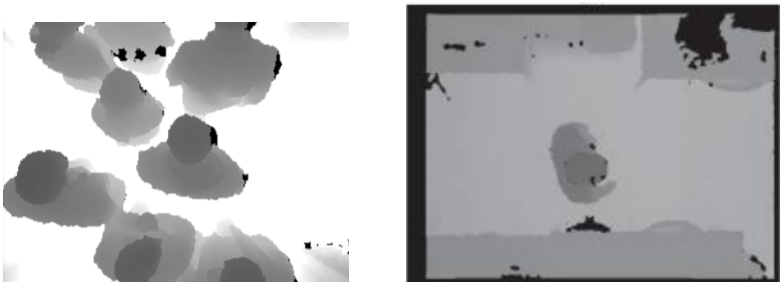
2.3.2. Depth Datasets

Using a time-of-flight sensor (ToF) or depth sensor such as a Microsoft Kinect v2 [88] [26], one can capture 3D positional information of the visible surfaces in a scene. Such sensors are not sensitive to environmental lighting conditions since unlike passive sensors that absorb environmental light, they are active sensors that emit infrared (IR) waves into the scene. These IR waves are then received back by an IR camera and a depth sensor calculates the distance of the point of reflection from the camera. Depth images are also preferred as they are color/texture invariant and provide us with person silhouettes that can be sufficient in terms of appearance details for human pose estimation. They resolve depth ambiguity although it is only applicable when scenes are captured from a side view.

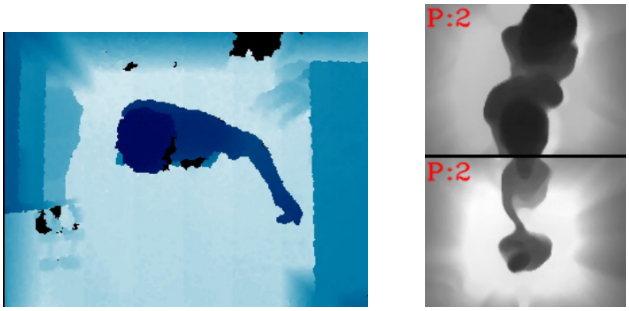
Although depth sensors are robust to environmental conditions, they produce noisy artifacts in the image and fail to provide range data at the edges of surfaces. This depth shadow is preprocessed using *hole-filling algorithms* such as bilateral filters [71]. These sensors are also currently only suitable for indoor environments and hence can capture only a limited scene depth.

Depth sensors have been used for tasks like hand gesture analysis, indoor mapping and object tracking but this report only focuses on datasets that were created for human activity analysis. One of the first open depth datasets for pose, the Stanford TOF dataset [19] and the Stanford EVAL dataset [20] captured depth sequences which were annotated by a motion capture system to produce 3D poses. With the release of commerical depth cameras like the Microsoft Kinect, there has been a proliferation of depth datasets. Motivated by the need for a large-scale corpus of depth images, the ITOP (Invariant Top-View) [27] dataset was released which used a 2-camera setup instead of motion capture system to capture 3D pose where one camera faces the subject and the other camera is located on the ceiling and pointed downwards at the subject. These datasets all involved single subjects which were facing a camera in a straight manner. Depth datasets can also be generated using a graphics simulator (e.g. Blender) and the Unicity [78] dataset adopts this method to generate synthetic images of human behavior inside an airlock. While the Unicity dataset uses virtual depth cameras placed on top of the scene, the DIH [51] dataset uses a side-view camera to capture scenes of multiple people.

There also exist other depth datasets like the TVHeads [45] for depth-based head semantic segmentation and TVPR [44] dataset for person re-identification. Samples of depth images can be seen in figure 2.14 and 2.15 for scenes captured from the top-view and front-view respectively.

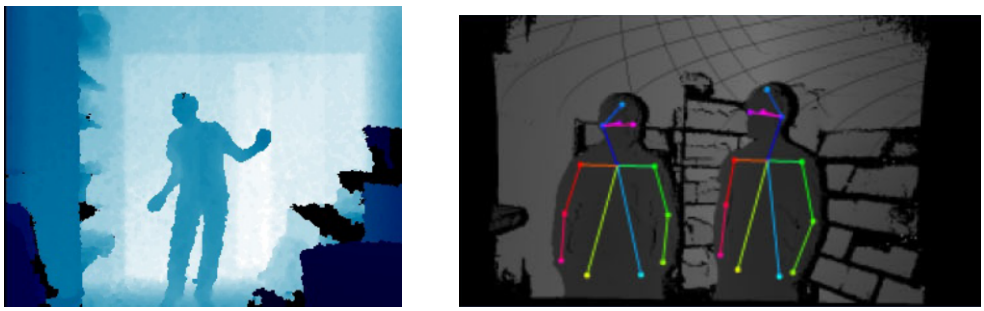


(a) TVHeads dataset [45] (b) TVPR dataset [44]



(c) ITOP dataset [27] (d) Unicity dataset [78]

Figure 2.14: Representative images from depth datasets captured from the top view.



(a) ITOP dataset [27] (b) DIH dataset [51]

Figure 2.15: Representative images from depth datasets captured from the front-view

2.4. Human Pose Metrics

Various metrics for pose estimation have been proposed that evaluate the localization accuracy of either the body joints or body parts.

Body-Part based metrics

- **PCP (Percentage of Correct Parts)** - If the predicted body joints at the ends of a body part segment are within 50% of the ground truth segment length, the body part made from these joints is considered accurate [16]. For 2D pose estimation, this metric shall require higher precision for body parts which are further from the camera but have similar size in the real world to body parts placed closer to the camera (i.e. *foreshortening*). Thus, this measure is relaxed with body parts closer to the camera and stricter with body parts placed further away from the camera. This measure also has a side effect of every body joint being evaluated by a different distance threshold depending on the size of the body part it belongs to.

Body-Joint based metrics

- **PCK (Percentage of Correct Keypoints)** - This metric defines a predicted body joint as accurate if the euclidean distance between the predicted and true joint is within a certain threshold. The original definition of the metric proposed by [84] considers this threshold to be a ratio of the size of the tightest box around a persons body joint annotations. An update version called PCKh [5], considers the threshold to instead be a fraction of the length of the head-neck body part segment. Note that this method does not penalize any false positives by the model.

- **PDJ (Percentage of Detected Joints)** - Similiar to PCK, this metric considers a predicted joint to be accurate if its euclidean distance from the ground truth joint is within a certain threshold of the torso diameter [75]. The torso diameter is defined as the distance between the left-shoulder and right-hip. In general since bodies with shorter limbs also posses smaller torsos, this metric evaluates the detections relative to the body size. This metric also does not penalize any false positives by the model and only measure the true positive rate.

- **mAP (Mean Average Precision)** - While the above metrics are usually used for single person pose estimation and also do not penalize false positives, mAP which was originally derived from object detection literature is more suitable for multi-person pose estimation. Using the predicted body-joints, poses are generated and a single pose is assigned to each ground truth pose based on the highest average PCKh. Average precision [84] for each body joint is calculated by using the precision-recall curve. The mean of these average precisions is reported as the mAP.

- **MPJPE (Mean Per Joint Position Error)** - This metric, usually used for 3D human pose estimation, calculates the per joint position error (PJPE) as the euclidean distance between the predicted joint and ground truth joint location. The mean error of all the joints is the final reported value. This is usually done after aligning the root joint (e.g pelvis) of the ground truth and prediction.

3

Method

The primary goal of this work is to extract accurate 3D human poses exhibited by the non-patient stakeholder(s) within the ICU experimental setting. The top-view of the experimental setting, presence of multiple non-patient stakeholders and variable spatial settings of a scene are outstanding challenges that need to be overcome. Finally, the proposed model should also be able to disambiguate the body joints of the patient from the non-patient stakeholders. To this end, a dataset is annotated which captures the variability of scenes of the experimental setting and a model is then built that is capable of understanding the diverse set of poses in these scenes.

3.1. Dataset

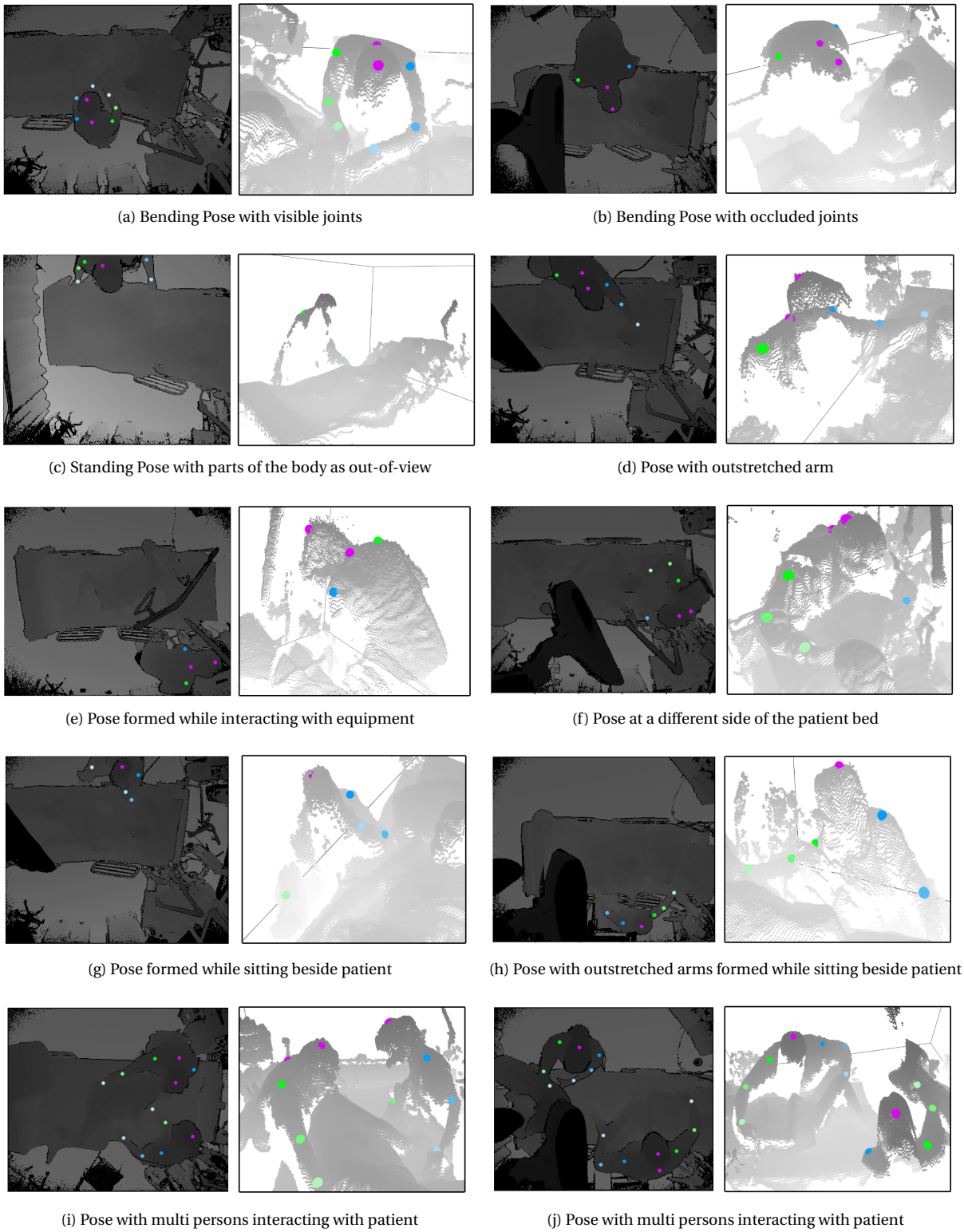
Extraction of 3D human pose in a clinical setting requires the collection and annotation of a dataset that contains a wide variety of activity types in diverse environmental setups. The experimental setting of this work involves multiple ICU rooms, each fit with a single Microsoft Kinect v2 camera on the ceiling. This camera captures a top-view scene of an ICU room using an infrared (IR) sensor with a resolution of 512 x 424 pixels and at a rate of 30 frames per second. The sensor is coupled along with a depth processor that outputs a depth image. The pixel values of the depth image indicate the distance (or depth) of a point in the real world with respect to the virtual plane of the sensor in millimeters(mm) . It should be noted here that the depth value is not the distance between a real world point and the sensor itself. The sensor outputs both IR and depth images with a datatype of *unsigned int16*. Although the theoretical value for a pixel in the depth image can range from 0-65535, most of the ICU rooms within the dataset have a maximum value never greater than 4000 (i.e 4000mm or 4m).

A typical ICU room would contain any combination of the following elements - patient, patient bed, medical equipment and other non-patient stakeholders such as nurses, doctors and visiting family. The presence of these elements leads to a highly variable spatial layout for the ICU rooms being recorded. In most scenes, the patient and their bed is the central focus of the scene with the non-patient stakeholders and medical equipment occupying the space surrounding the patient bed. The focus is on capturing those scenes where the non-patient stakeholders are interacting with the patient since such activities output a wide range of poses. Depth images are used as they preserve the privacy of individuals being monitored.

The dataset contains annotations for 8 upper-body joints i.e. 1)*head*, 2)*neck*, 3)*shoulder-right*, 4)*elbow-right*, 5)*wrist-right*, 6)*shoulder-left*, 7)*elbow-left* and 8)*wrist-left* as seen in figure 2.1. The annotations are done on the IR image and can be translated to the depth image as they have the same coordinate system. We annotated 13 sequences across 9 patients which combined together contain 1830 depth frames. These annotations provide real world 3D coordinates of the body-joints with the depth sensor being the origin as seen in the different poses in figure 3.1 using a color coding described in figure 3.2. Since the annotations are performed on 2D IR frames [1], occluded body joints could not be annotated and hence every frame does not contain all the eight body joints. In various scenarios, there is also a loss of annotations since the depth map contains no information on depth for certain pixels, especially at the edges of objects as seen in figure 3.3. Looking at the distribution of joints in figure 3.4, it is noticed that the *head* has the highest number of

[1] Detailed annotation guidelines can be found in the Appendix

annotations since scenes are captured from a top-view. It is then closely followed by *shoulder-right, shoulder-left, wrist-right* and *wrist-left* which have slightly less number of annotations due to the bodies of the non-patient stakeholders falling out of the field of view. Finally, the *neck,elbow-right* and *elbow-left* have the least number of annotations due to intra-body occlusions.



(a) Bending Pose with visible joints

(b) Bending Pose with occluded joints

(c) Standing Pose with parts of the body as out-of-view

(d) Pose with outstretched arm

(e) Pose formed while interacting with equipment

(f) Pose at a different side of the patient bed

(g) Pose formed while sitting beside patient

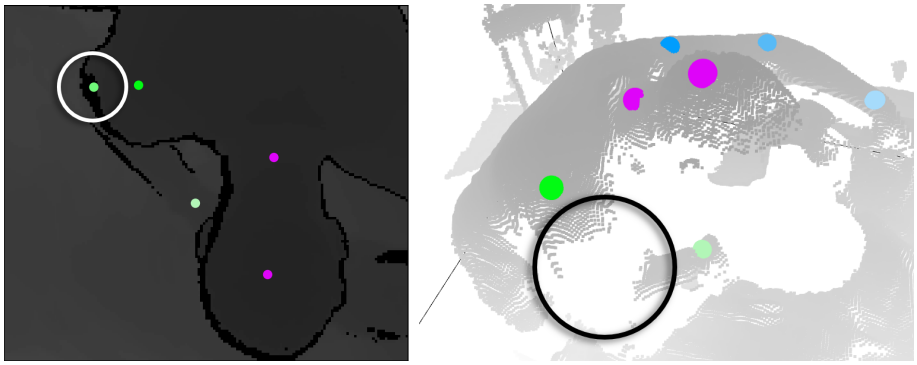
(h) Pose with outstretched arms formed while sitting beside patient

(i) Pose with multi persons interacting with patient

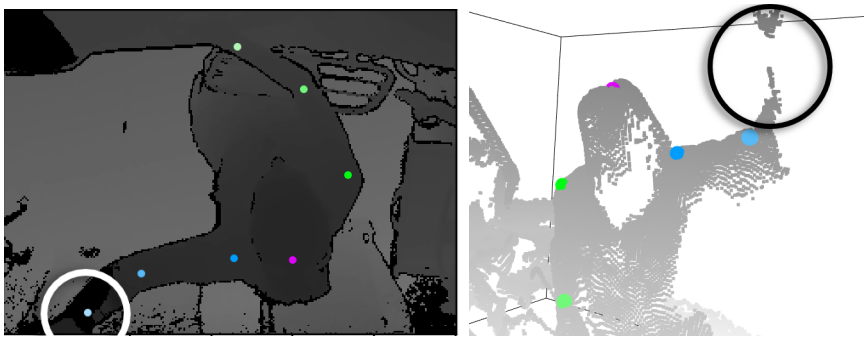
(j) Pose with multi persons interacting with patient

Figure 3.1: Different pose-types in the dataset shown using the IR image and truncated 3D point cloud with annotations.

Figure 3.2: Color coding used for the different upper body joints



a) Missed annotation for *elbow-right*



b) Missed annotation for *wrist-left*

Figure 3.3: The figure shows the depth image and corresponding 3D point cloud with annotations. Circled annotations are those that were discarded due to lack of depth information. A lack of information is represented with black pixels in the depth image and hence a lack of 3D point clouds.

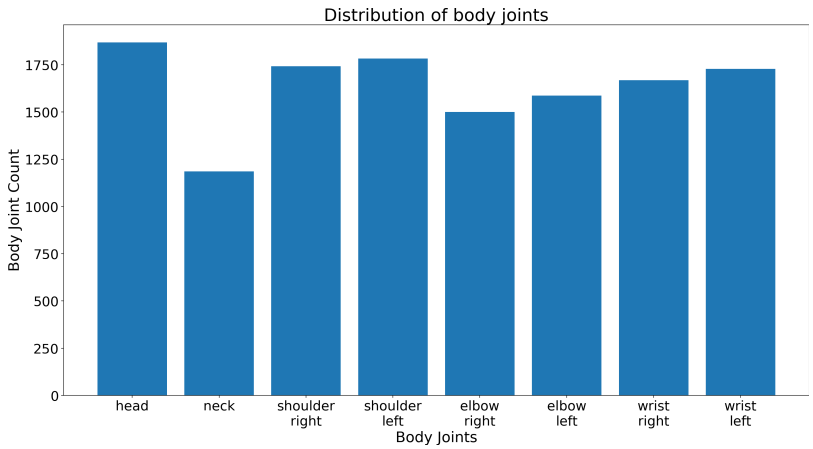
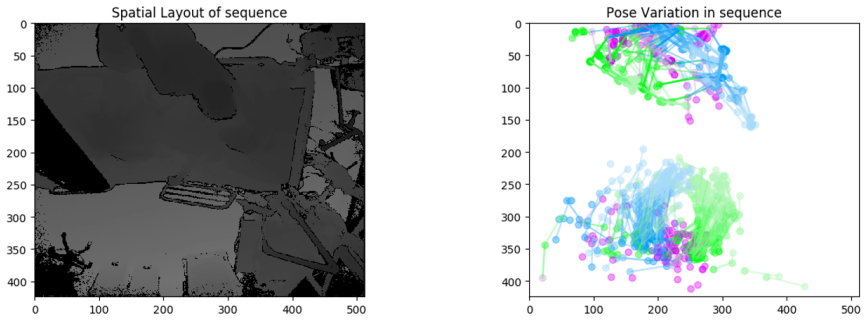


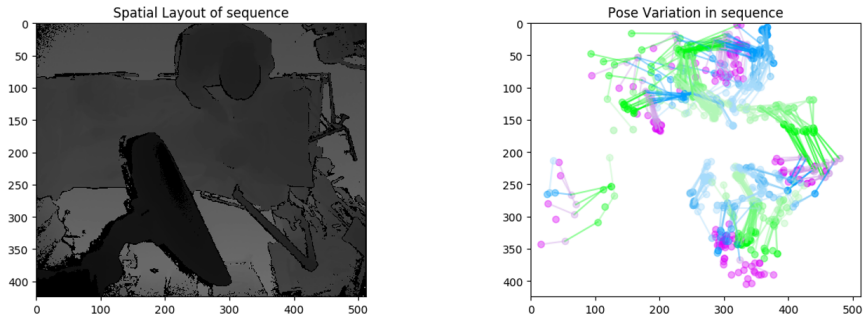
Figure 3.4: Class distribution of body-joints

Pose Variation

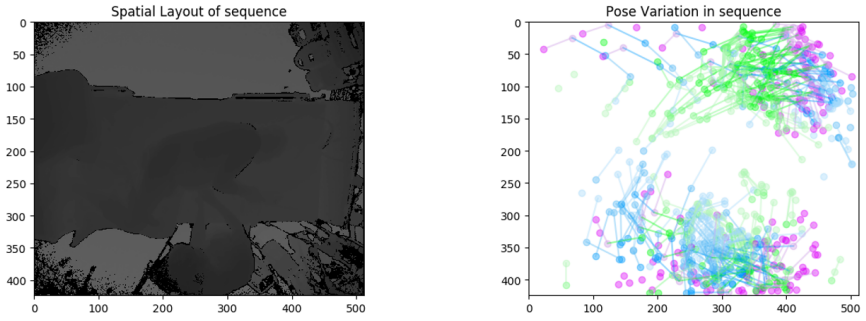
A concern to be kept in mind while building a pose dataset is variability in exhibited poses. Another matter relevant to the experimental setting in this work is also the spatial spread of various poses throughout an ICU scene. Ensuring this provides a form of data augmentation since similar poses can now be found in multiple locations and across different spatial settings. Such a study was done using figures shown in figure 3.5



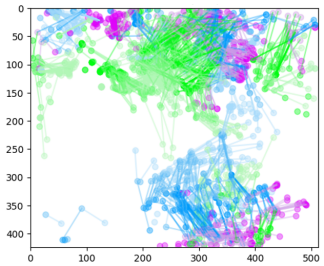
(a) Sequence 1



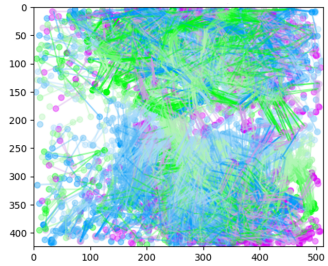
(b) Sequence 2



(c) Sequence 3



(d) Test Sequences



(e) Train Sequences

Figure 3.5: The figure [a),b),c)] shows the spatial layout and pose variation in sample sequences. In d) and e) one can also see the pose variation across the entire test and train subset.

3.2. Baseline Model

Having surveyed literature on human pose estimation using depth data, the baseline model is chosen as a 3D Convolutional Neural Network. This CNN [77] is a 3D extension of the OpenPose [7],[8] method, which requires as input a 3D voxelized point cloud and outputs voxel-wise probability heatmaps for the body joints and body parts connecting them. These heatmaps are then parsed (i.e. post-processed) to extract the human pose of the non-patient stakeholder(s).

Motivation

The openly available ITOP dataset [27] is closest to the scenarios from the ICU experimental setting of this work in terms of the image capture technology and the camera viewpoint. It is much simpler in terms of the spatial layout of the scene with just simple furniture resembling a table and a shelf. It only captures a single person at any given moment with a depth camera of spatial resolution lower than the Microsoft Kinect used in this work. This dataset has been benchmarked by a wide variety of algorithms as seen in table 2.1. The top two results on the ITOP leaderboard are both 3D CNNs but with a difference in their neural architectures. The method placed second [53] is an encoder-decoder architecture that only provides body-joint heatmaps and does not provide any information on body-joint parsing to create human poses. The method placed first and the method used as a baseline [77] in this approach uses a multi-head neural architecture to output both body-joint and body-part heatmaps which can be combined to output human pose when multiple persons are present in a scene.

3.2.1. Input and Output

Input

For the input, a 2D depth frame from the Kinect is converted to a 3D point cloud since the human bodies in it shall contain spatial structures in real-world coordinates (metres). To validate if the body joint annotations done on the 2D image provide equivalent body part lengths across frames, a box plot is created as shown in figure 3.6. Thus, the model shall not have to learn any perspective distortions due to 2D imaging artifacts [53]. Due to each ICU room having its own spatial constraints, the camera's central axis is not perpendicular to the ceiling, but slightly titled. To avoid any additional learning of such an affine transformation, the 3D point cloud is rotated such that its ground plane is parallel to the x-y plane of the camera coordinate system. To reduce computational complexity, all 3D points below the patient bed are removed since the task is to estimate the upper body joints and almost every scene in the dataset has the non-patient stakeholders in some form of a standing pose. This rotated 3D point cloud is then further processed into a binary occupancy grid. The original 3D point cloud is unordered in nature and hence cannot be consumed by the baseline CNN which needs its input to be in a lattice structure. Thus, the 3D point cloud needs to be discretized by converting it into a voxel grid with a fixed voxel resolution. Under the binary occupancy mechanism, if a voxel contains more than one 3D point, it is marked as 1, otherwise it is marked as 0. The annotated body joints are converted to 3D voxel points using the same process. A pictorial representation of this process can be seen in step 1 and 2 of figure 3.9.

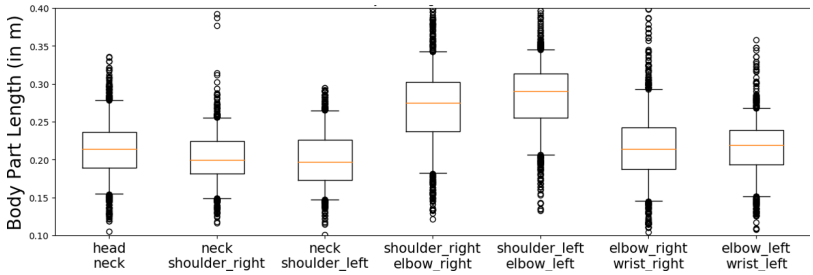
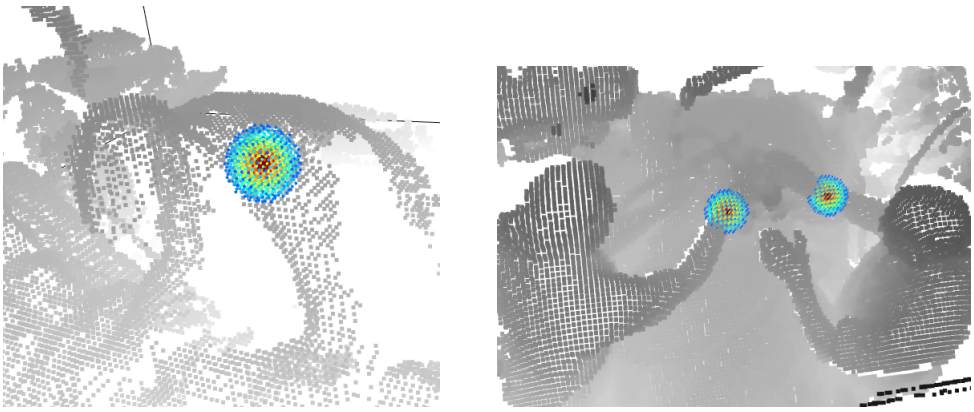


Figure 3.6: A box plot showing the distrubution of body-part lengths across frames. Due to annotation errors, some frames have body part lengths which are outliers.

Output

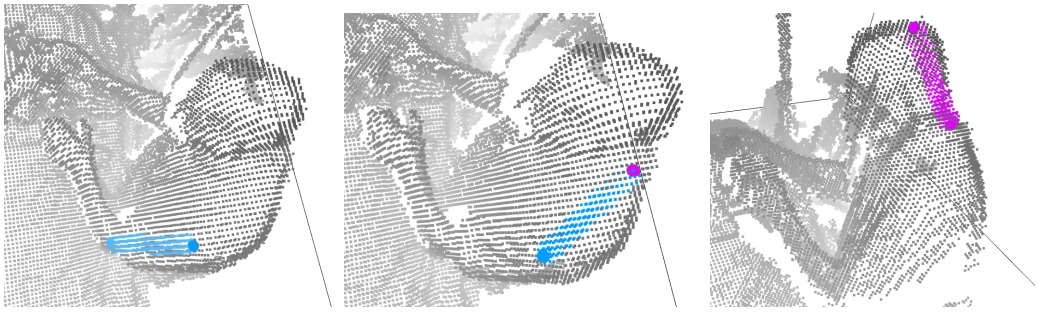
The baseline network outputs two sets of heatmap grids - 1) a body joint heatmap set and 2) a body part heatmap set, each containing probability values. A set of voxel grids, each containing gaussians of small variance and centered on the true 3D location of a joint serves as one ground truth output heatmap (J). A

sample body joint heatmap can be seen in figure 3.7. Conceptually, heatmaps were chosen as regression targets instead of direct 3D coordinates, since they are a representation of the confidence in the location of a body joint. The body part heatmap (B) is a collection of voxel grids, with each containing 3D cylinders indicating the body parts that connect a pair of body joints. The line segment between the two body joints forms the axis of the ground truth 3D cylinder as seen in figure 3.8. The voxel values inside this cylinder are marked as 1 while values outside it are marked as 0. The model benefits by having knowledge of the 3D location of the body parts as it allows for a simple greedy parsing of the body-joints into a body-joint-connected human pose.



(a) *shoulder-left* (b) *wrist-right*

Figure 3.7: 3D Gaussian heatmaps for different body joints shown as being centered around the ground truth location. The rainbow colormap has been used to show the confidence values of each voxel.



(a) *shoulder-left - elbow-left* (b) *neck - shoulder-left* (c) *neck - shoulder-left*

Figure 3.8: 3D cylindrical heatmaps for different body parts. The value of each voxel in the ground truth body part heatmap is equal to 1.

3.2.2. Network Architecture

The network architecture, as seen in figure 3.10, consists of two primary heatmap regression sub-networks – 1) a feature extraction sub-network (in blue) that attempts to understand the structure of the input scene and 2) a task-oriented sub-network (in pink and green) that uses the voxel features for human pose estimation. As is common in deep neural nets, a feature extraction stage uses a sequence of 3D convolutional kernels to output a set of feature maps (F). The baseline method applies 3 convolutional blocks such that each block performs 3D convolutions of (7 x 7 x 7), (5 x 5 x 5) and (3 x 3 x 3) respectively. Each convolutional block contains a 3D convolution operation followed by a max-pooling, ReLU [54] non-linearity and batch-normalization operation [31]. As the input is processed deeper into the network, the spatial dimensions of the convolutional kernels are decreased, but the number of activation maps is doubled. This is a standard approach in most CNNs where the number of representations (i.e. activation maps) of a visual input are increased as one goes deeper and the size of the convolutional kernels is reduced so as to not cause a computational overload.

The output of the feature extraction sub-network (F) is consumed by the task-oriented sub-network that is further decomposed into two parts – 1) a skeletal body-joint estimation head (in pink) and a 2) bodypart

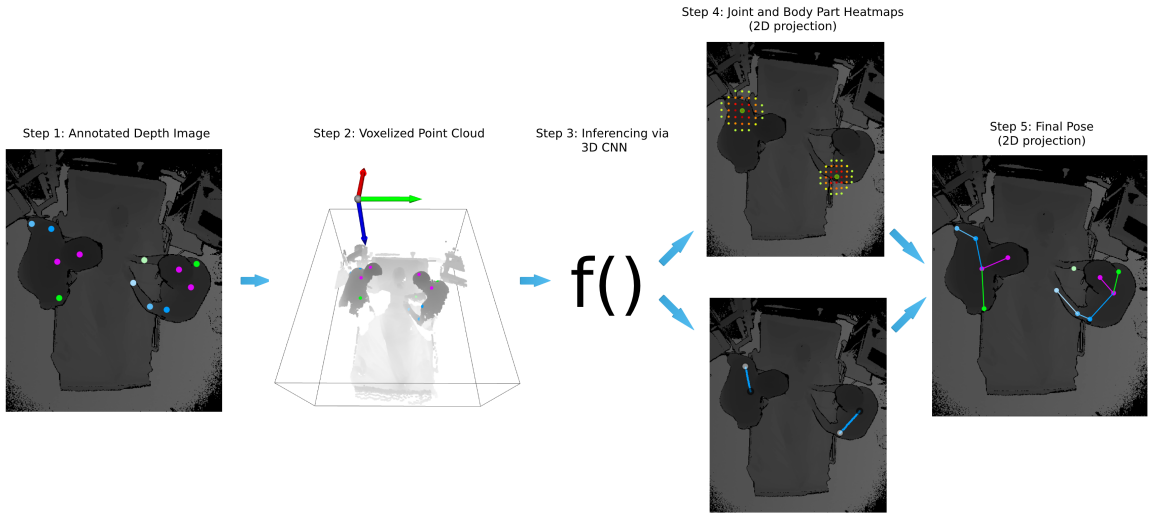


Figure 3.9: Modeling pipeline showing the different stages of [77]. In Steps 1 and 2, 2D images are annotated and all 2D information is projected to 3D. In Step 3, the 3D CNN *f()* network consumes the 3D point cloud and produces output heatmaps. The output heatmaps in Step 4 are parsed to produce a final pose in Step 5.

estimation head (in green). They then perform another series of convolutions on the feature map F to estimate the skeletal body-joint heatmaps (J) and the body-part heatmaps (B). The task sub-network may be repeated multiple times (t=[1, ...,T) as shown in figure 3.10 to improve output quality. The input to subsequent iterations of the sub-networks is an activation map which is a concatenation of the feature extraction maps along with the body joint and body parts heatmaps. Such a multi-representation input to the successive sub-networks may guide the network to pay attention to the salient parts of the scene (i.e the body joints and body parts). It also helps the network to form latent representations that better understand the relationships between various body joints and body parts as referenced by the kinematic chain of the human body in figure 2.1. The body joint and body part estimation heads have the same neural architecture which is three convolutions of (3 x 3 x 3) followed by two convolutions of (1 x 1 x 1). Convolutions of (1x 1 x 1) are a computationally cheap way to summarize and reduce the number of activation maps since the output activation maps need to be equivalent to the number of body joints and body parts for each of the task-oriented sub-networks respectively. Note, that the number of activation maps stays the same across both these sub-networks except at the final convolutional block. The final convolutional block does not contain a non-linearity for the purpose of loss calculation.

Nonlinearity and Batch Norm

A standard convolutional block in the feature extractor of the baseline model consists of a convolutional operation followed by a max-pooling operation. The block then ends with a batch-normalization [31] and ReLU [54] layer. The batchnorm layer has been shown to improve the training process by making the process more stable [64]. The ReLU layer adds non-linearity to the neural network and has been preferred over other non-linear functions such as sigmoid or tanh from empirical observations. There has been a disparity in the order of applying the batch-norm and ReLU layers across different works, with no theoretical justifications for which comes first. Thus, this report relies on previous implementations and applies BatchNorm followed by ReLU since this is not the core focus of this work.

3.2.3. Model Training

Loss Function

To train the network, a selective L2-loss function applied at each stage (t=[1, ...,T), yields losses f_J^t , f_B^t for the joint-estimation head and body-part estimation head respectively. A selective loss implies that we do not penalize a joint if it is unavailable as ground truth since a top-view prohibits annotation of self-occluded joints. A pictorial representation of the selective L2-loss can be found in figure 3.11. Note, that a loss is applied at the end of every stage t for intermediate supervision of the model. Using a stochastic gradient optimizer,

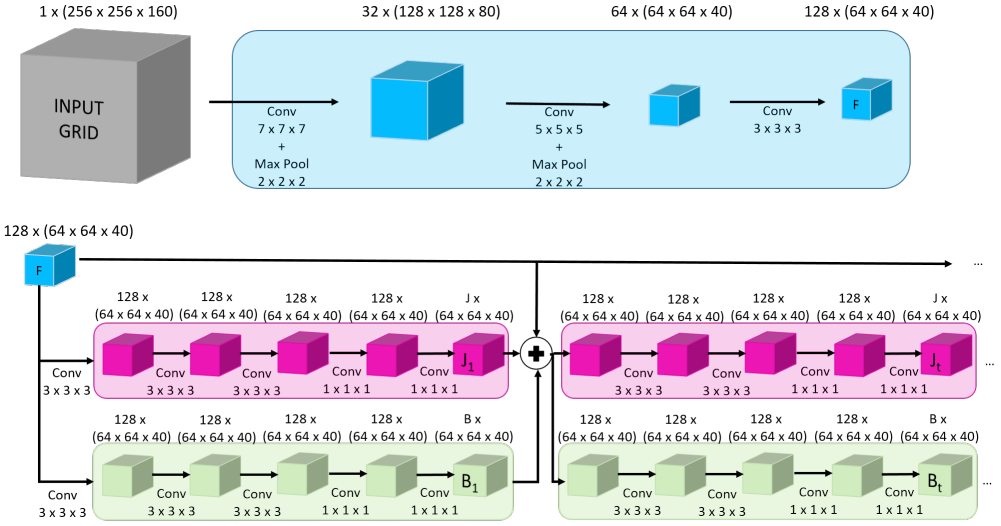


Figure 3.10: The 3D CNN network of [77] with its three primary components is shown – feature extraction F (in blue), joint estimation J (in pink) and body part estimation B (in green). The + sign indicates a concatenation which serves as an input for the subsequent iterations of the joint estimation and body part estimation networks.

we update the neural network weights using a fixed learning rate and the following objective function:-

$$f = \sum_{t=1}^{T} (f_J^t + f_B^t) \tag{3.1}$$

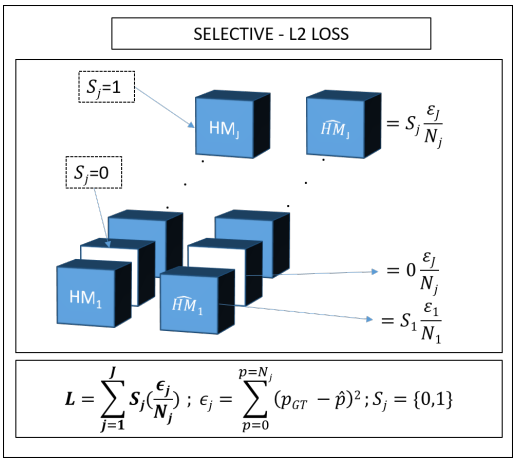


Figure 3.11: The selective L2-loss does not penalize heatmaps for which there is no ground truth. Here $S_j = \{1,0\}$ represents whether a particular heatmap is selected or not. ϵ_j refers to the squared error between a ground-truth heatmap HM$_j$ and predicted heatmap $\hat{HM}_j$ and N_j refers to the total number of voxels in a heatmap. The final loss L is a sum of the individual heatmap errors.

The above loss function was not used in the original method of [77] since they use a dataset which has all ground truth data available. Since it has not been possible to annotate all joints given the experimental setting of a top-view camera, a selective loss function was chosen.

Post-Processing

The network outputs J body-joint heatmaps for each of the J annotated body joints and B body-part heatmaps for the body-parts formed by the body joints as per the kinematic chain. A fixed confidence threshold is applied on these heatmaps before any further post-processing steps. The predicted body-joint heatmap(s) are parsed via non-maximal suppression (NMS) to extract 3D location of the body joint(s). Non-maximal suppression finds the local maxima in a predefined region by comparing a dilated and original heatmap.

The dilated heatmap is made by a morphological operation which uses a fixed shape called the structuring element to find the "bright" regions in matrix. Thus, it enables one to find regions of high confidence in a heatmap. A NMS threshold ensures that two peaks in the heatmap are always separated by a fixed distance. Thus, the gaussian peak(s) in the predicted body-joint 3D heatmaps are captured by the NMS approach. To parse (i.e. connect) the predicted body-joints, we refer the kinematic chain of the human body (e.g. shoulder-left is followed by elbow-left) and find that specific combination of body joints with maximum predicted body-part heatmap values between them.

Although this model is only trained for eight upper-body joints, this choice can be application dependent. If one is able to annotate the lower body joints accurately, the baseline model is capable of learning and predicting these joints as well.

Stopping Criterion

As learning proceeds, a criterion has to be defined to stop the learning process. A standard approach when training a machine learning model is to mark that epoch when the validation metric reaches its peak and is on a downward slope. Since this report compares the baseline model with other proposed models, it is decided to stop the training after a fixed number of epochs so that each model has an equivalent exposure to the number of training samples. This stopping epoch is chosen as that point where the baseline model sees a drop in its validation metric.

3.3. Proposed Model

The baseline model suffers from pose estimation issues such as poor localization of body joints (*false positives*), missed predictions (*false negatives*), confusion between left and right body joints and spurious detections on non-human surfaces(*false positives*). The methods described below hope to abate these issues with variations made to the network architecture.

3.3.1. Increased Receptive Field

The feature extraction model of the baseline model performs a simple series of convolutions using a standard convolutional block [2]. It is possible to improve the model performance by increasing its receptive field by additional convolutional layers as seen in figure 3.12.

Motivation This approach has been motivated by observing the decrease in classification error with increase in neural net layers on the ImageNet dataset [13] as seen in figure 3.13. One of the reasons why convolutional neural networks are able to perform well on vision based tasks is the increase in the receptive field of each neuron as more convolutional layers are added. A diagramatic explanation of this has been provided in figure 2.5. Additional layers of convolutions have the capability to increase the receptive field of the baseline network, and hence provide more context to the network. Additional layers also lead to increased non-linearity and a deeper hierarchy of feature combinations. The use of a richer representation at the end of the feature extractor is also useful during the iterative refinement stages of the baseline method.

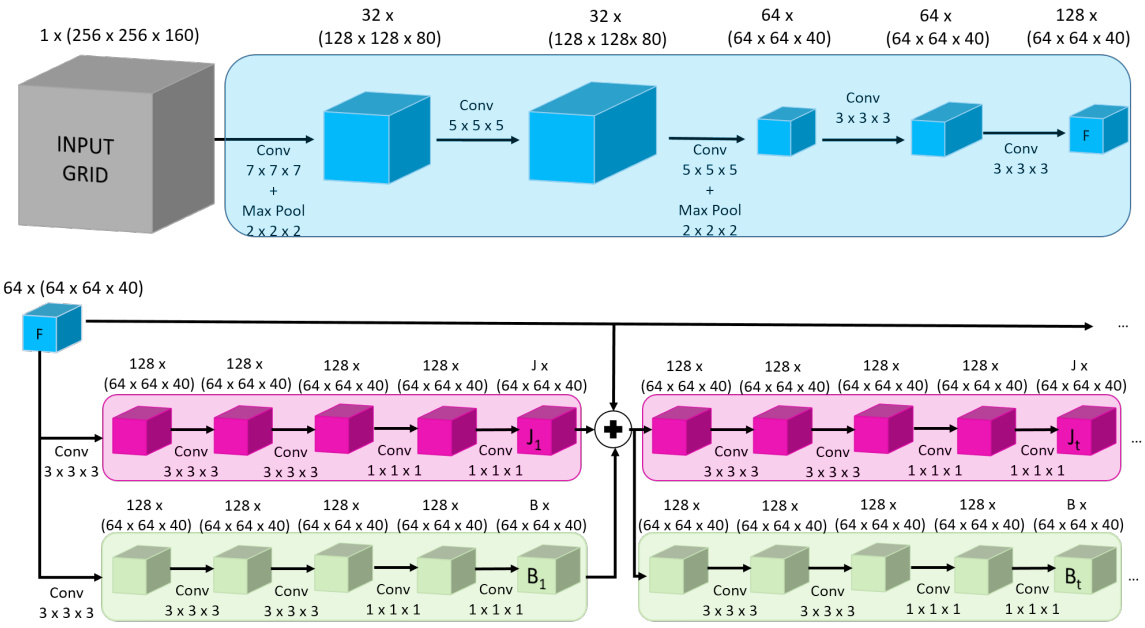


Figure 3.12: Improved feature extraction using additional convolutional blocks in the baseline model. Dimensions of the input, output and intermediate activation maps are displayed outside the colored boxes. The operations and their respective kernels sizes are displayed inside the colored boxes alongside their respective arrows.

Architecture The baseline feature extractor now receives an additional processing on certain output activation maps. Specifically, adding a (5 x 5 x 5) and (3 x 3 x 3) convolutional operation without any max-pooling to maintain voxel grid dimensions, allows for improved understanding of the input scenes by enhanced feature extraction. Although more convolutional operations can be added in either the feature extractor or the task sub-networks, it leads to incremental GPU memory usage. The addition of activation maps in the feature extractor leads to a memory usage cost due to the presence of intermediate activation maps. Thus, the neural architecture has bounds on how "*deep*" it can be made due to hardware constraints.

[2] A convolutional block typically consists of a convolutional layer followed by a max-pooling and non-linear activation layer.

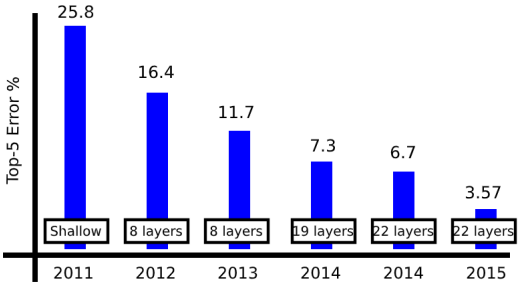


Figure 3.13: Neural net layer count along with the error metric shown for the winners of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [13] challenge in different years.

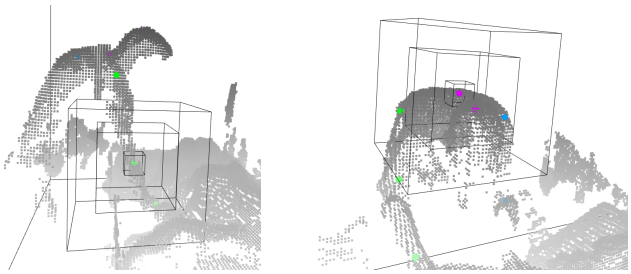
Receptive Field Analysis A detailed analysis of the receptive field at each layer of the network has been done in table 3.1 for the baseline model and in table 3.2 for the model with increased receptive field. The rows highlighted in blue indicate the feature extraction layers. One can notice that for a sample input resolution of (256,256,160) the feature extractor of the baseline model has a receptive field of (26,26,26) while the proposed model has a receptive field of (42,42,42). The expanded receptive field as shown in figure 3.14 may help the model establish more spatial context enabling it to better localize body joints and avoid common issues with human pose estimation such as left-right confusion.

Input	Operation	Output	Receptive Field
(256,256,160)	Conv3D (7,7,7)	(256,256,160)	(7,7,7)
(256,256,160)	MaxPool3D(2,2,2)	(128,128,80)	(8,8,8)
(128,128,80)	Conv3D(5,5,5)	(128,128,80)	(16,16,16)
(128,128,80)	MaxPool3D(2,2,2)	(64,64,40)	(18,18,18)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(26,26,26)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(34,34,34)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(42,42,42)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(50,50,50)
(64,64,40)	Conv3D(1,1,1)	(64,64,40)	(50,50,50)
(64,64,40)	Conv3D(1,1,1)	(64,64,40)	(50,50,50)

Table 3.1: Analysis of the receptive field of the baseline model. The colored text represents the operatiosn of the feature extractor.

Input	Operation	Output	Receptive Field
(256,256,160)	Conv3D (7,7,7)	(256,256,160)	(7,7,7)
(256,256,160)	MaxPool3D(2,2,2)	(128,128,80)	(8,8,8)
(128,128,80)	Conv3D(5,5,5)	(128,128,80)	(16,16,16)
(128,128,80)	Conv3D(5,5,5)	(128,128,80)	(24,24,24)
(128,128,80)	MaxPool3D(2,2,2)	(64,64,40)	(26,26,26)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(34,34,34)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(42,42,42)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(50,50,50)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(58,58,58)
(64,64,40)	Conv3D(3,3,3)	(64,64,40)	(66,66,66)
(64,64,40)	Conv3D(1,1,1)	(64,64,40)	(66,66,66)
(64,64,40)	Conv3D(1,1,1)	(64,64,40)	(66,66,66)

Table 3.2: Analysis of the receptive field of the proposed model. The colored text represents the operations of the proposed feature extractor.



(a) *elbow-left* (b) *neck*

Figure 3.14: Snapshots of 3D point clouds showing (7x7x7),(24x24x24) and (42x42x42) bounding boxes centered around a joint

3.3.2. Upsampling in Feature Extractors

To further work on improving the receptive field of the feature extractor, one can also perform convolutions at different resolutions and merge the resulting activations maps as shown in figure 3.15. This section shall explain the motivation behind such an approach.

Motivation Processing of both coarse and fine information and eventually merging them together may lead to a more richer feature representation of the input scene. Such an approach has been adopted in human pose estimation literature wherein information from multiple stages with different resolutions is concatenated to successfully improve the performance of the model [55], [10]. This is done since the activation maps at course scales contains information aggregated over large spatial distances. One method to merge coarse and fine information would be to perform the same set of feature extraction operations on raw voxelized inputs of different resolutions. Another method would be to use internal feature maps and perform operations on their coarse and fine representations. It is chosen to perform convolutional operations on coarser feature maps instead, since they contain relevant information on the salient regions of the network as well. To conclude, this method employs the same concept as that of a feature pyramid, an approach used in the early days of computer vision for multi-scale feature representation.

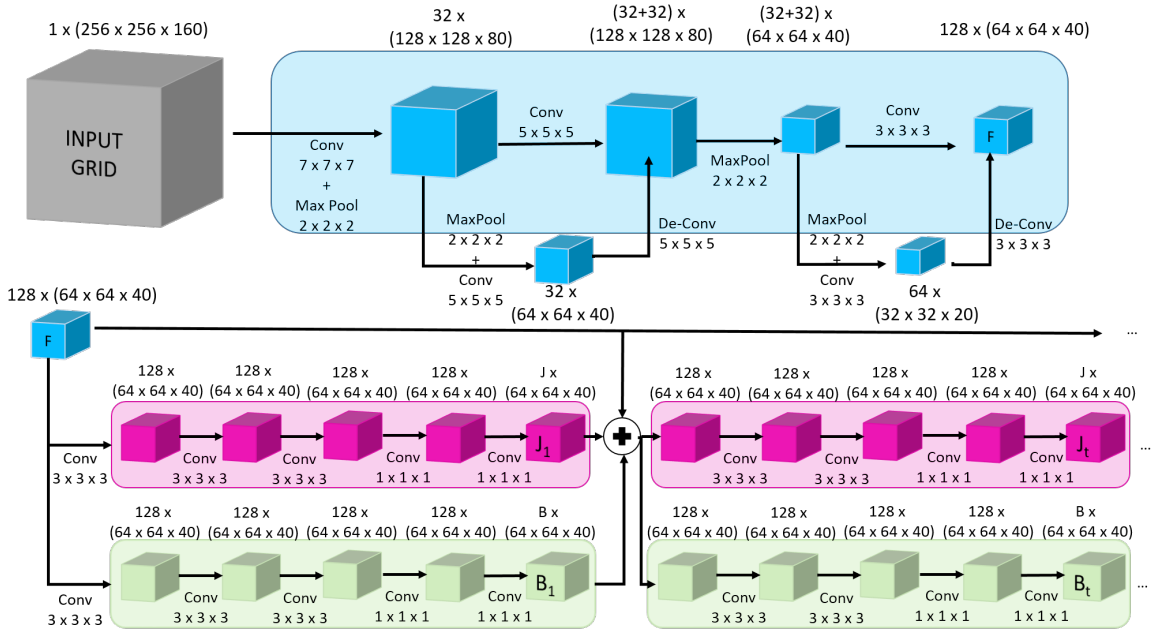


Figure 3.15: The feature extraction module (in blue) contains additional convolutional operations in the form of downsampling-upsampling sub-networks.

Architecture After the initial convolutional operation of (7 x 7 x 7), a two stream sub-network is created. While one branch performs a (5 x 5 x 5) convolution at the same dimension, the other branch downsamples the activation map, perform a (5 x 5 x 5) convolution and then performs a learned upsampling (*De-Conv*) to the original dimension. A similiar set of operations is then applied at the next stage using a (3 x 3 x 3) convolution. Thus, the model is able to perform the same type of convolutional operation at two different dimensions - i.e a coarse and a fine dimension.

3.3.3. Sequential Sub-Networks

The baseline model performs convolutions in the body joint and body part sub-networks in parallel, while this section explains a neural architecture in which the network perform sequential operations in the sub-task networks. The network first attempts to understand the body parts and then proceeds to extract the body joints as seen in figure 3.16.

Motivation In a multi-person setting, the joints of multiple persons have to be parsed into their individual poses. If one is given a set of body joints in 3D space, without any information, it may not be a simple task to parse these joints into individual poses. However, it is much simpler to identify the locations of body joints given a body part segment in 3D. Such an approach was directly inspired from [8] where they perform a similar operation in 2D. Intermediate supervision at the end of each body-part or body-joint sub-network allows the model to improve the intermediate output heatmaps and hence possibly leverage spatial dependencies.

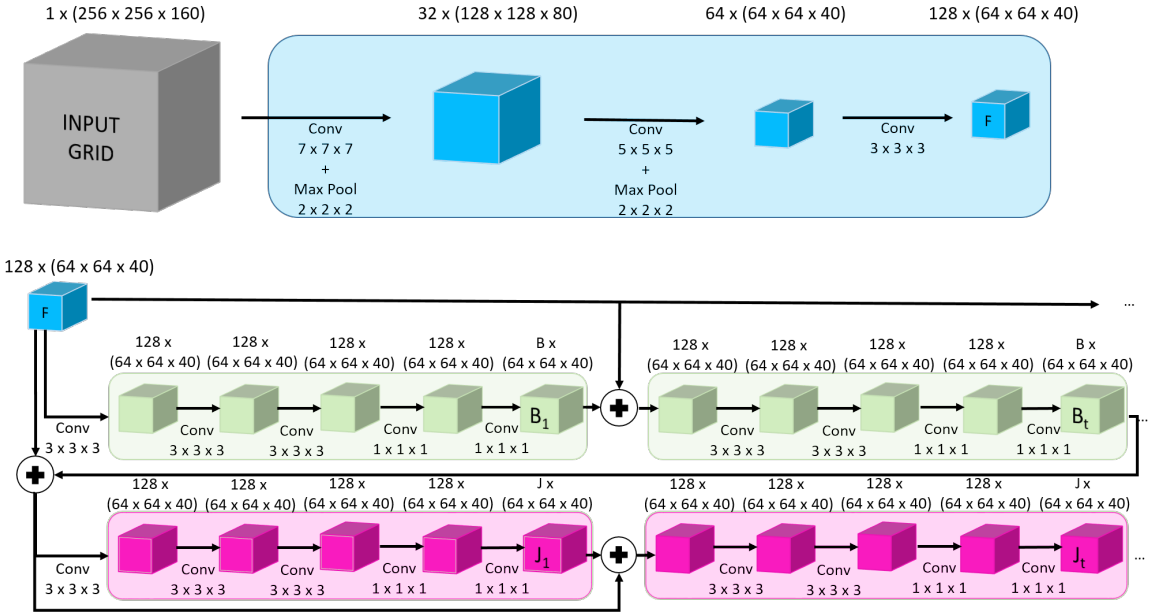


Figure 3.16: Usage of sequential data pipeline, instead of a parallel data pipeline in the body joint and body part sub-networks. Data first flows through the body part sub-network (in green) and is then processed in the body-joint networks (in pink)

Architecture After the feature extraction module, the network perform a series of body-part heatmap regressions. The last body-part heatmap along with the feature extractor is passed to the first body-joint heatmap regressor. Subsequent body-joint heatmap regressors takes the last body-part heatmap and feature extractor as additional inputs. Thus, during inference, the network first predicts body-part heatmaps, and uses them as contextual information to regress body-joint heatmaps.

4

Results

This section shall present the performance of the different neural architectures as discussed in sections 3.2 and 3.3. The results are presented by evaluating upon different sequences from the annotated dataset described in section 3.1. The performance of the baseline model is reported in Section 4.3 and the performance of the proposals are reported in Section 4.4. The evaluation metrics and implementation details are described in sections 4.1 and 4.2 respectively.

4.1. Evaluation Metrics

This chapter uses the standard precision-recall and F1-score curves to evaluate a model. The confidence threshold applied on the body-joint and body-part heatmaps are varied from 0.3 to 0.9 with increments of 0.1. The thresholded heatmaps are then parsed to output the 3D joint coordinates and the connections between them. The definition of precision and recall is as follows:-

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

In equation 4.1 and 4.2, TP, FP and FN are True Positives, False Positives and False Negatives respectively. A predicted joint is considered a TP if it is within the TP radius threshold as shown by the shaded region in figure 4.1 . This can be visualized as a 3D sphere centered around the ground truth 3D location of a body joint as shown in figure 4.2. A predicted joint is considered a FP when it is outside the ground truth sphere. When no prediction is made for a ground truth joint, it is considered as a FN prediction. This work evaluates and compares models using a TP radius of 10 cm as it is a standard practice in works solving for human pose estimation. This is done so that the performance of models on the dataset used in this work can be at least qualitatively compared to other open datasets [77].

While evaluating the model performance on an annotated test datasest, a predicted joint has to be associated with the ground truth joint of a specific person. A 3D bounding box with padding is constructed around each annotated person using the ground truth annotations. If a prediction falls inside a persons bounding box, it is assigned to that person. If a prediction is not assigned to any person, it considered a FP. If a prediction inside a bounding-box has no ground-truth available, it is simply ignored. A detailed algorithm is explained in appendix section A.

Comparison Methodology

Following the standard practice of cross-validation, two datasets are created from the thirteen annotated sequences with the training subset of each dataset containing ten sequences and the testing subset containing three sequences. The testing subset of each dataset contains sequences from different patients to evaluate for generalization capability. The test subset of each dataset contains patient sequences that are not available in the train subset as would be the case in a real-world deployment of the model. Dataset 1 consists of a train-test split of 1414-248 while dataset 2 consists of a train-test split of 1315-265.

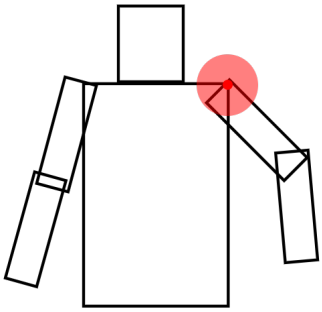
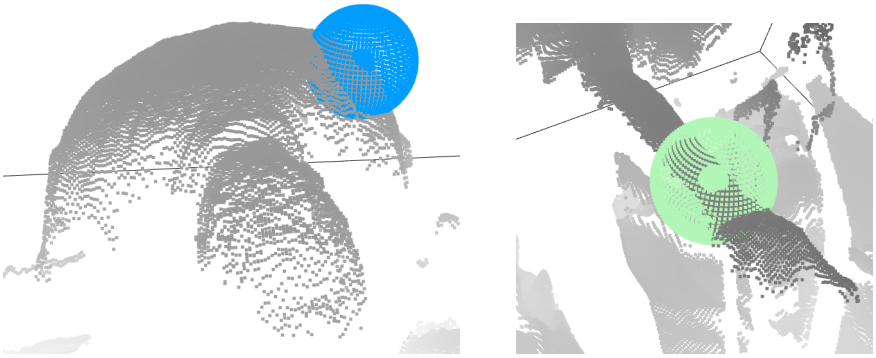


Figure 4.1: This figure shows a 2D illustration of the TP region. The dark red point is the ground truth annotation of the *shoulder-right* joint. The TP region is a spherical region represented by the light red circular area in the image.



(a) *shoulder-left* (b) *wrist-right*

Figure 4.2: Visualization of the true positive region (radius=10 cm) in the unordered 3D point cloud

The relative ordering of the performance of each model across both these datasets shall provide insight into the true performance of each model. Mean and standard deviation of each models performance on a particular dataset have not been reported due to high training time requirements. It is likely that a favorable initialization of the weights of the 3D CNN may help improve the performance of one model over another. However, this work fixes the random seed so as to avoid such ambiguities. This random seed controls the initialization of neural network weights as well as the order in which a neural network is exposed to training samples.

4.2. Implementation Details

This section discusses the process of generating the train and test datasets as well as the training and inference procedures.

Data Preparation As mentioned in section 3.2, the input 3D point cloud is unordered in nature and needs to be voxelized. A voxel resolution of 1.2 cm was chosen instead of 2.8 cm as done in the baseline method [77]. This resolution was chosen from a range of 1.0 cm to 2.0 cm to gain maximum resolution and also allow the baseline model to fit into the GPU memory. Post voxelization, every sequence undergoes a fixed affine transformation to make its ground plane parallel to the plane of Microsoft Kinect. Once transformed, 3D points within a fixed threshold from the ground plane are removed so that only point close to and above the patient bed are part of the voxel grid used as input to the neural network. The final dimension of the input grid is (256 x 256 x 160) which has the same range in the x and y axes but reduced dimension in the z-axis since 3D points below the patient bed are removed. All 3D point clouds were rotated by a 180 degrees for the purpose of data augmentation. The output heatmaps are generated at a one-fourth resolution of the input voxel grids. The body-part gaussian heatmaps have a standard deviation of 3 and the body-part cylindrical heatmaps have a radius of 2 voxels.

Training Procedure The neural nets and the selective L2-loss function were implemented using the PyTorch framework [56]. The random seed generator was fixed (=42) to ensure reproducibility of results and the datatypes of all tensors was fixed to a single precision floating point format (i.e. *float32*). Since the output heatmaps are quite sparse in nature a learning rate of 0.3 was chosen for the stochastic gradient optimizer to maintain acceptable training speed. A batch size of 2 is chosen to ensure that the backpropagated gradients are always averaged across frames. The choice of frames in a particular batch is randomized at each epoch. A higher batch size could not be chosen due to GPU memory constraints. The count of training epochs is chosen by referring the training procedure of the baseline model. As shown in figure 4.3, the test errors flattens at epoch 20 and shows no improvement henceforth.

Inference Prodecure Since the output heatmaps are generated at a resolution which is one-forth that of the input grids, they are upsampled to the original resolution via trilinear interpolation. A heatmap confidence threshold of 0.6 is chosen to evaluate the metrics of the model. The NMS distance threshold is fixed to 10 voxels (=12cm) which ensures that two local peaks shall always be separated by that distance. This value was chosen since the true positive threshold is 10cm and it would be desirable to have only one local peak in that region.

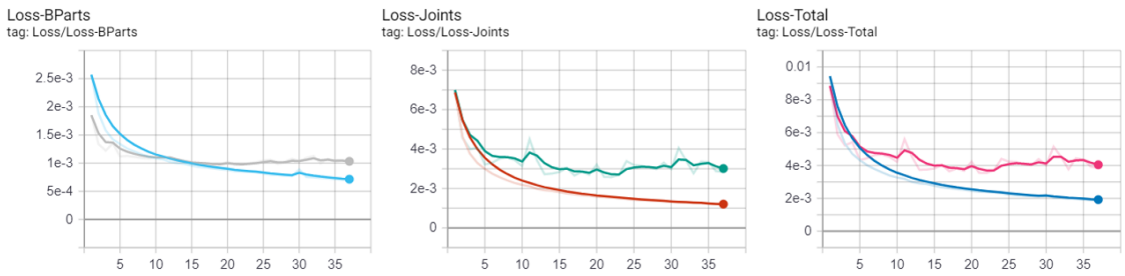


Figure 4.3: The graphs above show the loss curves for the body-joint, body part and their total (left to right). The light blue, red and dark blue curves represent the training loss, while the grey, green and pink curves represent the test loss.

4.3. Baseline Model

The baseline model as described in section 3.2 is benchmarked on dataset 1 and dataset 2 to evaluate its performance. The goal is to evaluate whether the chosen baseline is capable of producing robust human poses from the top-view ICU scenes captured in the dataset.

4.3.1. Single Stage Baseline

The single-stage baseline model performs computations for the body-part and body-joint sub-networks just once and in parallel to yield human pose. The motivation behind this experiment is to evaluate the performance of the baseline model in its most basic form and understand the main classes of incorrect predictions made by the model. These are - confusion between the left and right sides of the body, missing body joints or disconnected body joints as per the kinematic chain and finally detections made on surfaces not belonging to the non-patient stakeholders (i.e. spurious detections)

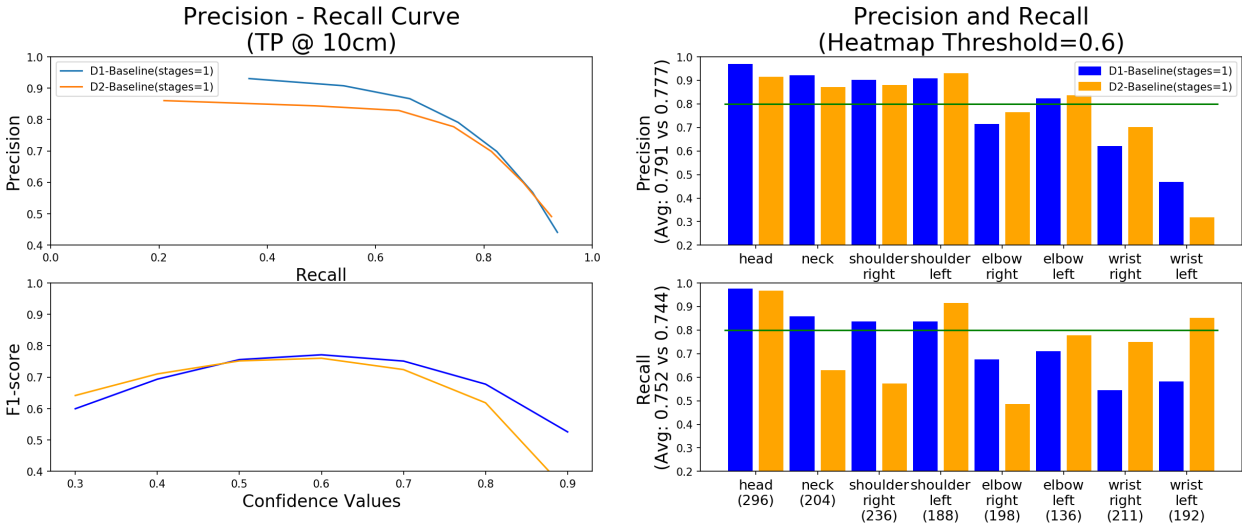


Figure 4.4: The left figure shows graphs for the precision-recall and F1-score curves. The right figure shows per joint precision and recall at a heatmap confidence of 0.6. The green line in the right figure indicates the minimum acceptable threshold for body-joint metrics. *D1-Baseline(stages=1)* refers to the single-stage baseline model benchamarked on dataset 1 while *D2-Baseline(stages=1)* refers to the same model for dataset 2.

Looking at the F1-curves for both dataset 1 and 2 in figure 4.4, it can be seen that the model performance peaks at a particular heatmap confidence threshold. This indicates that at lower values of heatmap confidence, the output heatmaps contain noisy voxels. The performance of the model again falls for higher values of heatmap confidence indicating that its confidence on the location of various body-joints is low. The individual body-joint metrics indicate that the model performance varies by a large margin between the two datasets. In dataset 1, the precision and recall metrics for the *head*, *neck* and *shoulder* joints are higher than those of the *elbow* and *wrist* joints. Thus, the performance deteriorates as one moves down the kinematic chain to the extremities of the human body. This could be due to the fact that the *elbow* and *wrist* joints are in many cases partially occluded and also display a wide range of 3D spatial structures. For dataset 2, there seems to be no particular trend visible in the precision and recall metrics of individual body joints.

The main classes of errors can be seen in figures 4.5 and 4.6 for dataset 1 and 2 respectively. A left-right confusion is caused due to lack of spatial context on the complete human pose and thus the single-stage baseline makes predictions based on local information. Missing body-joints may be caused due to a lack of understanding on the spatial structures around a particular body joint. Finally, spurious detections are caused due to the model being unable to disambiguate between a body-joint and environmental clutter.

It can be seen in figure 4.7 that the model can at times mistake the patient's *wrist* body-joint as belonging to the non-patient stakeholders of the scene. This has been shown via a 2D projection of the ground truth and predicted heatmap grids which shows the confidence of the model in context of a particular joint. Location and similarity in 3D structure of the *wrist* body-joint could be possible reasons for such an output.

Figure 4.8 shows the model is able to roughly estimate the position of joints that have not been annotated due to visibility issues.

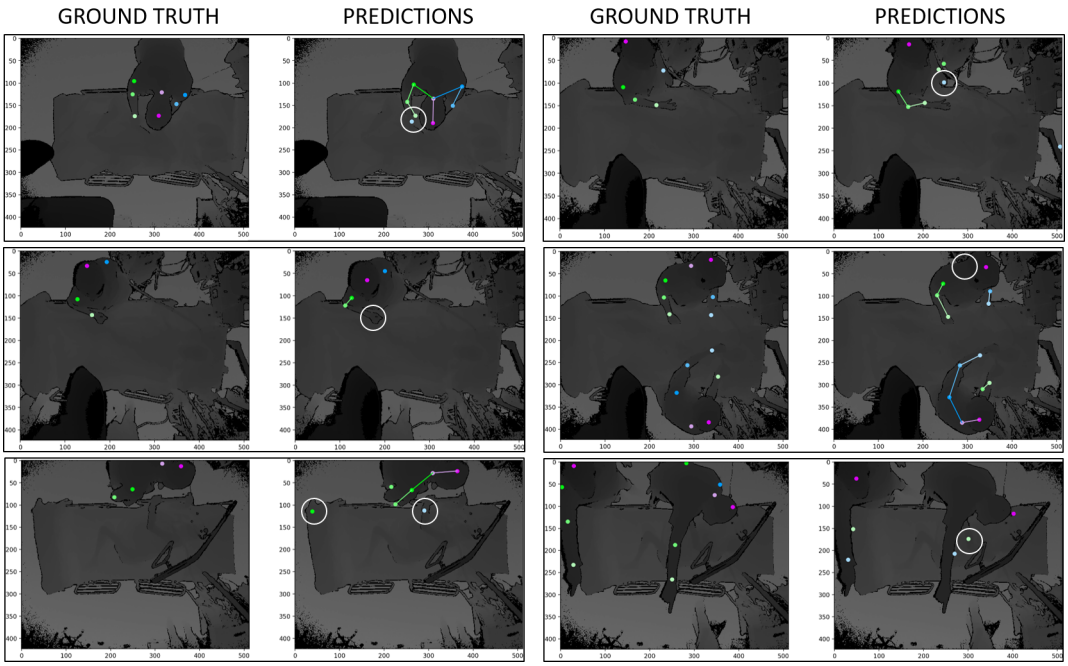


Figure 4.5: The images in the figure indicate the models performance on the various human poses found in dataset 1. It can be seen that the model suffers from left-right confusion (row 1), missing joints or missing body-parts (row 2) and spurious detections (row 3). The erroneous regions are denoted by a white circle.

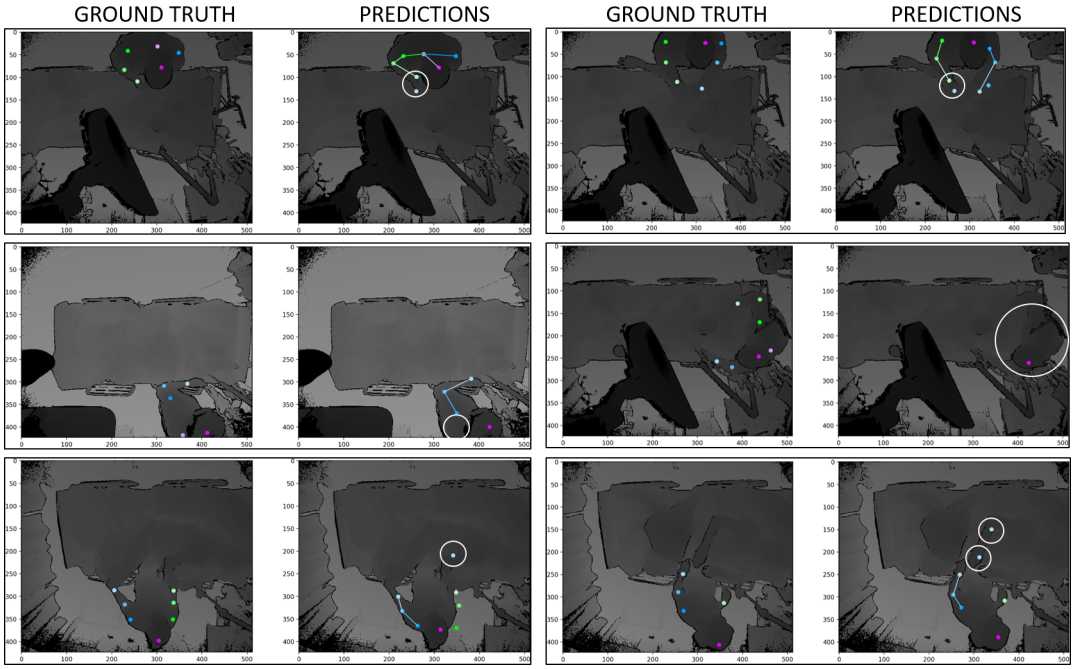
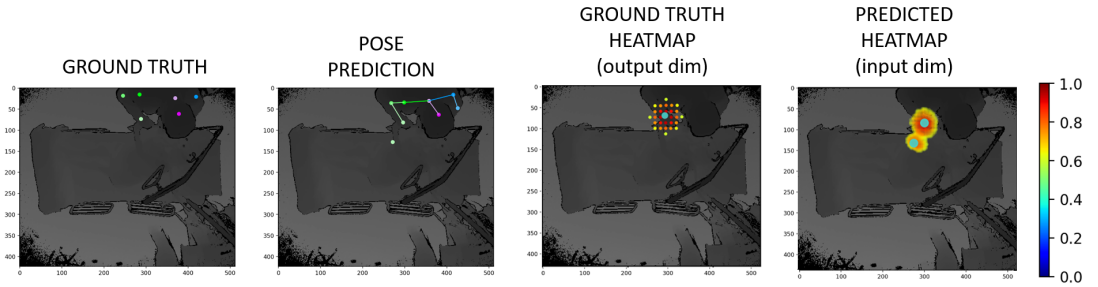
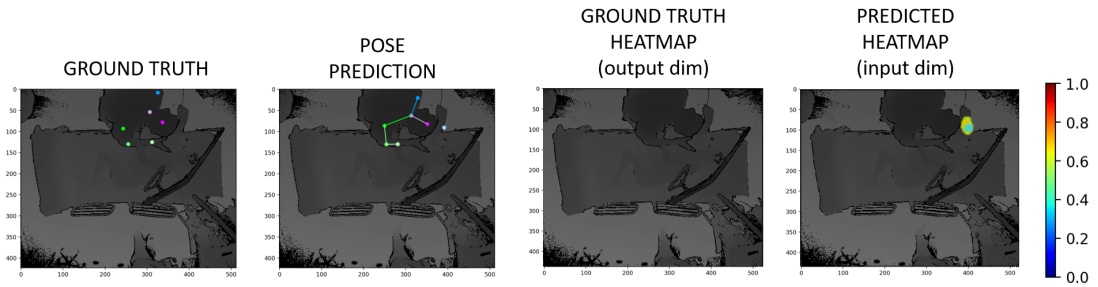


Figure 4.6: Sample predictions from dataset 2 showing that the model is capable of producing coherent poses for many frames but with various kinds of errors. The first row shows the left-right confusion, the second row shows missing joints and the third row shows spurious detections. The erroneous regions are denoted by a white circle.

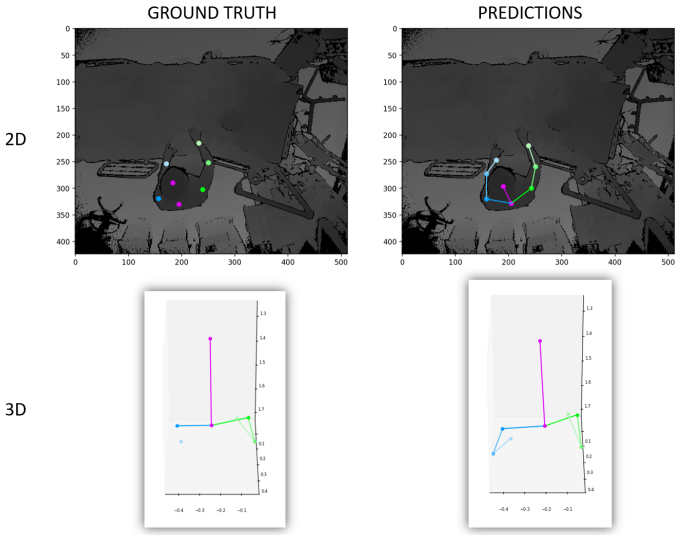


(a) *wrist-left*



(b) *wrist-left*

Figure 4.7: The figures above show the grouth truth annotation, pose predictions and the associated heatmaps for a specific joint. Note, the 2D projection of ground truth heatmap is sparser in nature compared to the prediction heatmap due to a difference in dimension of the source 3D heatmaps.



(a)

Figure 4.8: Ground truth and model predictions shown as projections on a 2D depth image and as points in 3D.

4.3.2. Multiple Stage Baseline

This section displays the results for the multi-stages baseline model with the goal to observe any improvements in comparison to the single-stage baseline. The motivation behind a multi-stage task sub-network is to provide subsequent stages with prior information on the possible locations of the various body joints and body parts.

The F1-score curves of dataset 1(figure 4.9) and dataset 2(figure 4.10) indicate an improvement for the multi-stage baseline compared to the single-stage baseline. The F1-score values also remain in the same range at lower values of heatmap thresholds showcasing a stability in the output heatmaps. There is also an improvement in the average recall across both datasets which indicates the fitness of a multi-stage model in detecting body-joints.

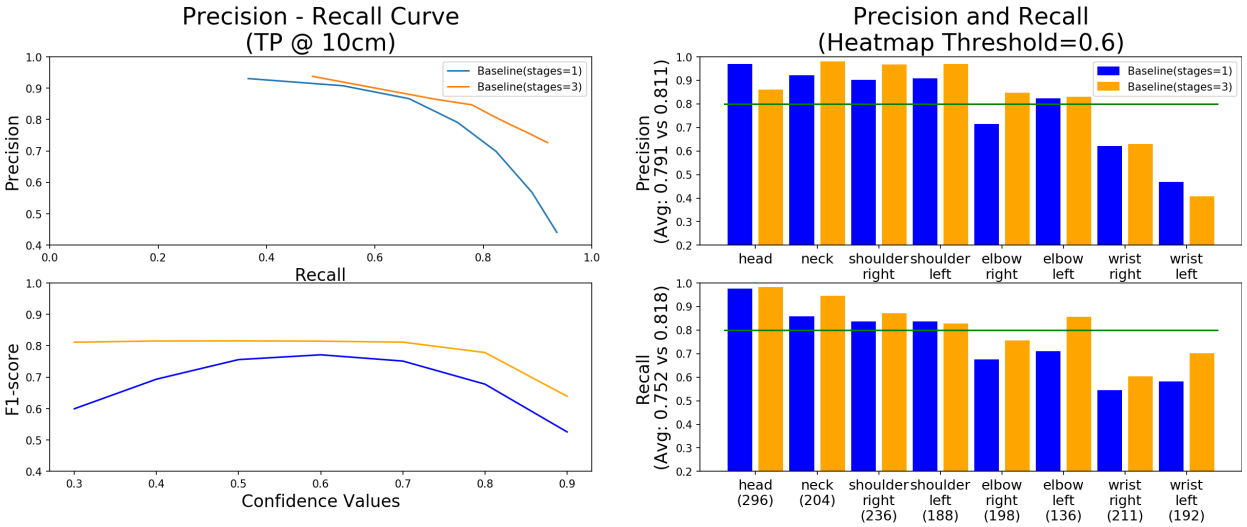


Figure 4.9: The F1-score of the multi-stage model on dataset 1 stays around the same value for lower confidence values indicating stability of the model in its predicted heatmaps

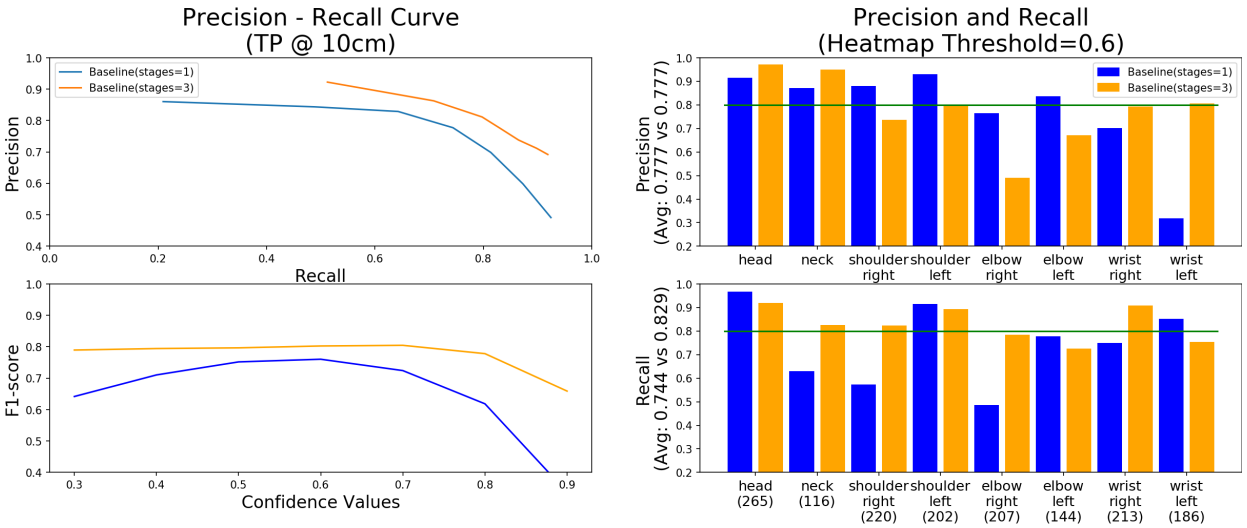
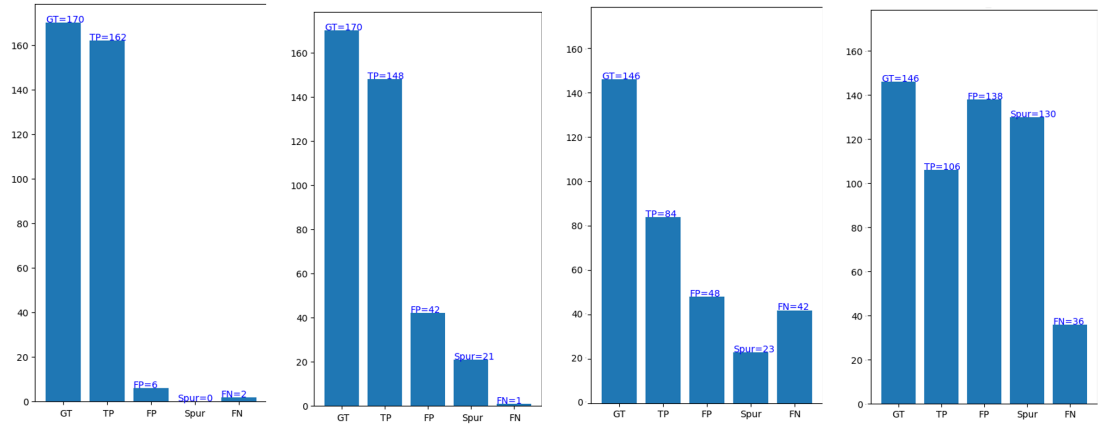


Figure 4.10: The F1-score curve of the multi-stage baseline on dataset 2 always hovers over the single-stage baseline indicating the average improvement in model performance.

Dataset 1

The *head* and *wrist-left* joints of the multi-stage baseline have a lower precision compared to the single-stage baseline due to the high number of spurious detections (i.e. false positives). Statistics for the same can be found in figure 4.11. Specifically, the *wrist-left* body joint has a high number of false positives due to incorrect predictions on a piece of thin medical equipment which the models mistakes as a body-joint. The reason for this repetitive error is unclear.

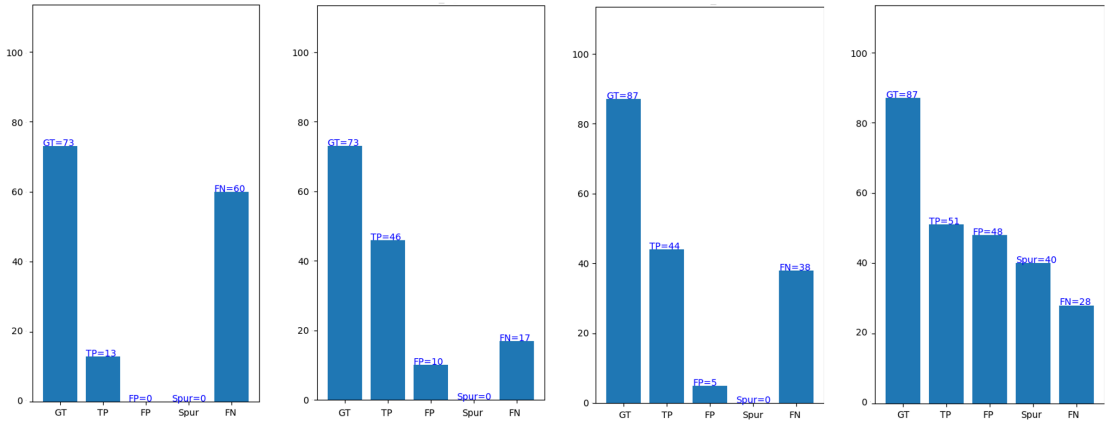


(a) Stage=1 (*head*) (b) Stage=3 (*head*) (c) Stage=1 (*wrist-left*) (d) Stage=3 (*wrist-left*)

Figure 4.11: The bar plots in this figure, indicate, for a specific joint, the Ground Truth(GT), True Positive (TP), False Positive(FP), Spurious(Spur) and False Negative(FN) count. The bar plots shown above compare the results between the single and multi-stage baseline for patient sequence 2 from the test subset of dataset 1.

Dataset 2

Recall of the *shoulder-right* and *elbow-right* body-joint improves by using the multi-stage baseline model. Although the recall of *elbow-right* body-joint increases, it suffers from a drop in precision as shown by the statistics in figure 4.12.



(a) Stage=1 (*shoulder-right*) (b) Stage=3 (*shoulder-right*) (c) Stage=1 (*elbow-right*) (d) Stage=3 (*elbow-right*)

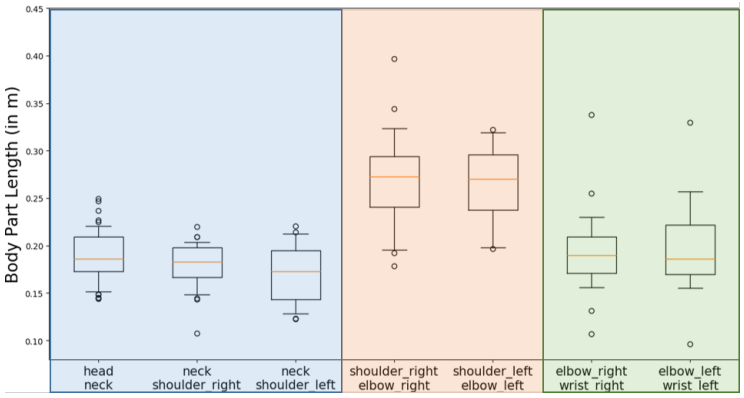
Figure 4.12: The bar plots in this figure, indicate, for a specific joint, the Ground Truth(GT), True Positive (TP), False Positive(FP), Spurious(Spur) and False Negative(FN) count. The bar plots shown above compare the results between the single and multi-stage baseline for patient sequence 2 from the test subset of dataset 2.

Computational Footprint An analysis of the computational complexity of the baseline model with different stages can be found in table 4.1. There is a major increase in the computational and parametric requirements for a multi-stage baseline, but this cost leads to a benefit in the performance of the model as seen in table 4.2.

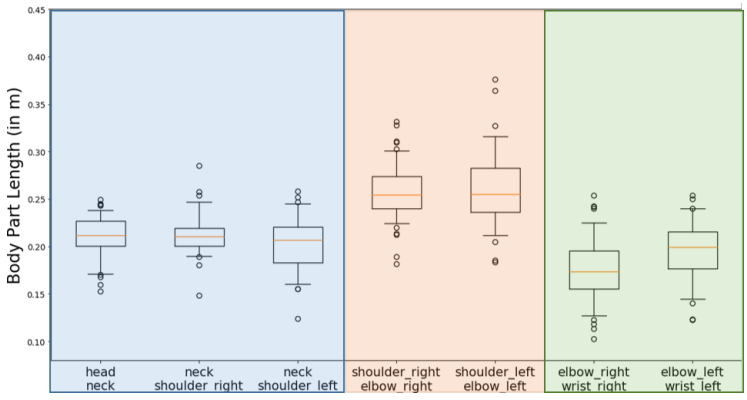
Model	Params Count	Params Size	Training Memory	Inferencing Memory
Baseline (stage=1)	3.2M	12.13 MB	~11 GB	~8.5 GB
Baseline (stage=3)	8.7M	33.44 MB	~13.9 GB	~11.5 GB

Table 4.1: Analysis of the baseline model with stages=1 and 3.

Predicted Body Part Lengths Figure 4.13 shows the different body part length distribution as predicted by the multi-stage baseline method. The color schemes in the figure illustrate different groups of body-part lenghts. The blue bar groups the *head-neck, neck-shoulder_right* and *neck-shoulder_left* body-parts since they have a median body-part length hovering around 20 cm. The orange bar groups the *shoulder_right-elbow_right* and *shoulder_left-elbow_left* body part since they have a median body-part length hovering between 25 cm and 30 cm. Finally, the *elbow_right-wrist_right* and *elbow_left-wrist_left* are grouped by a green bar since their body-part median lengths are in the 20cm range. These ranges match closely to the ranges seen for the ground truth body parts as seen in figure 3.6.



(a) Dataset 1



(b) Dataset 2

Figure 4.13: Different color bars group the body-parts based on their predicted median length.

4.3.3. Multiple Stage Baseline - Supplemental Studies

This section performs additional analysis on the multi-stage baseline to gain further insight into the models behavior. The results for the paragraphs below are summarised in table 4.2 for both datasets 1 and 2.

Augmentation

The models in subsection 4.3.2 are trained with a rotation augmentation where the 3D scene is rotated by 180 degrees. This section explores whether the addition of another affine augmentation shall improve the performance of the multi-stage baseline model. Specifically, the 3D point cloud is translated randomly along

the y-axis of the Microsoft Kinect camera. Although the space of affine transformations is quite large, only this transformation is chosen since the goal is to test the learning capacity of the baseline model. It can be seen from table 4.2 that the row with the model named as *Baseline-Aug* boosts the average performance of the multi-stage baseline to yield the best performing model. Although the augmentation described here improves model performance, it has not been used in other models due to an increase in the training time.

Optimizer
Neural networks are essentially optimization problems framed within a particular data structure. Optimizers attempt to update the weights of this data structure by minimizing the value of the specified loss function. The experiments above use the stochastic gradient (SGD) optimizer which is regular gradient descent but on mini-batches of the dataset. Another optimizer called Adam [39], has been shown to perform better due to its use of an adaptive learning rate by relying on the history of gradients. The goal of this experiment is to understand if the model, which is named *Baseline-Adam* behaves differently under different optimization schemes. The results in table 4.2 indicate that the *Baseline-Adam* model performs worse by a few points compared to the baseline model optimized by SGD. Thus, the SGD optimizer is chosen as the default optimizer for all experiments.

Reduced Supervision
To study the performance of the model with reduced supervision, an experiment is performed on both datasets where sequences are removed to reduce the size of the training corpus. The datasets are created in a manner that attempts to maintain the nature of the body-joint distribution within all datasets.

While dataset 1 (*D1.0*) consists of 1412 training frames, its reduced versions consist of 1166 (*D1.1*) and 943 (*D1.2*) frames respectively. Dataset 2 (*D2.0*) consists of 1360 training frames and its reduces versions consist of 1137 (*D2.1*) and 912 (*D2.2*) frames respectively. The results for models of dataset 1 named *Baseline-D1.1* and *Baseline-D1.2* show that while *Baseline-D1.2* performs worse, there is a slight improvement on using *Baseline-D1.1*. The models for dataset 2 show a trend of decreasing performance with *Baseline-D2.2* performing worse than *Baseline-D2.1*.

Modified TP threshold The TP threshold of 10 cm may not be accurate enough for a model to be used in real-world deployments. Hence, an analysis of the model is done using lower TP thresholds i.e. 8 cm, 6 cm and 4 cm. The models show decreasing performance as it to be expected with stricter evaluation thresholds.

Model	TP Threshold	Dataset 1			Dataset 2		
		Avg Precision	Avg Recall	Avg F1-score	Avg Precision	Avg Recall	Avg F1-score
Baseline (stages=1)	10 cm	0.791	0.752	0.771	0.823	0.668	0.738
Baseline (stages=3)	10 cm	0.811	0.818	0.815	0.777	0.829	0.802
Baseline-Aug(stages=3)	10 cm	0.819	0.848	0.834	0.832	0.819	0.825
Baseline-Adam (stages=3)	10 cm	0.784	0.804	0.794	0.769	0.777	0.770
Baseline-Dx.1 (stages=3)	10 cm	0.873	0.781	0.824	0.861	0.715	0.781
Baseline-Dx.2 (stages=3)	10 cm	0.761	0.801	0.781	0.790	0.644	0.710
Baseline (stages=3)	8 cm	0.760	0.809	0.784	0.726	0.820	0.770
Baseline (stages=3)	6 cm	0.599	0.775	0.675	0.568	0.780	0.657
Baseline (stages=3)	4 cm	0.331	0.670	0.443	0.292	0.646	0.402

Table 4.2: Human pose estimation metrics for datasets 1 and 2 on different versions of the baseline model. The *x* in the model named *Baseline-Dx* refer to the dataset number.

4.4. Proposed Models

An analysis of the different proposals from section 3.3 is done in this section. The comparisons between the baseline and proposed model are made on both datasets to better understand generalization capabilities.

4.4.1. Increased Receptive Field

This section utilizes the model from section 3.3.1 which uses a feature extraction module with additional convolutional operations and is referred to as *Baseline-Feat++* for the remainder of this report. The goal here is to analyze if the increase in receptive field helps generate a more informative feature hierarchy at the end of the feature extraction module.

The metrics for the *Baseline-Feat++* model can be found in figures 4.14 and 4.15.

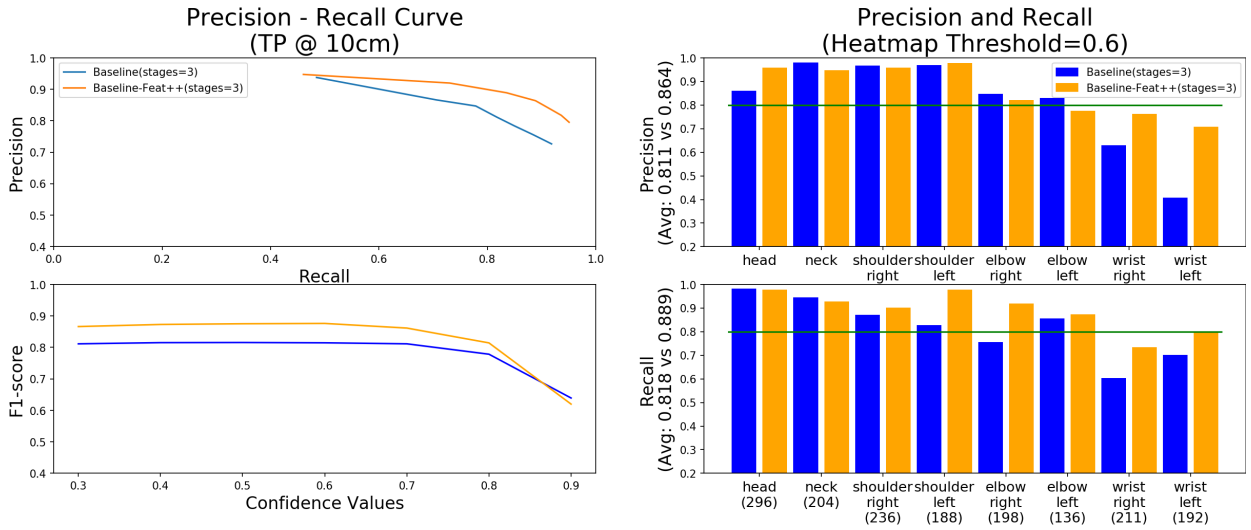


Figure 4.14: The F1-score curve shown in this figure shows an improvement over the multi-stage baseline model. Particularly, the *wrist* body-joints have increased recall and precision compared to the baseline model.

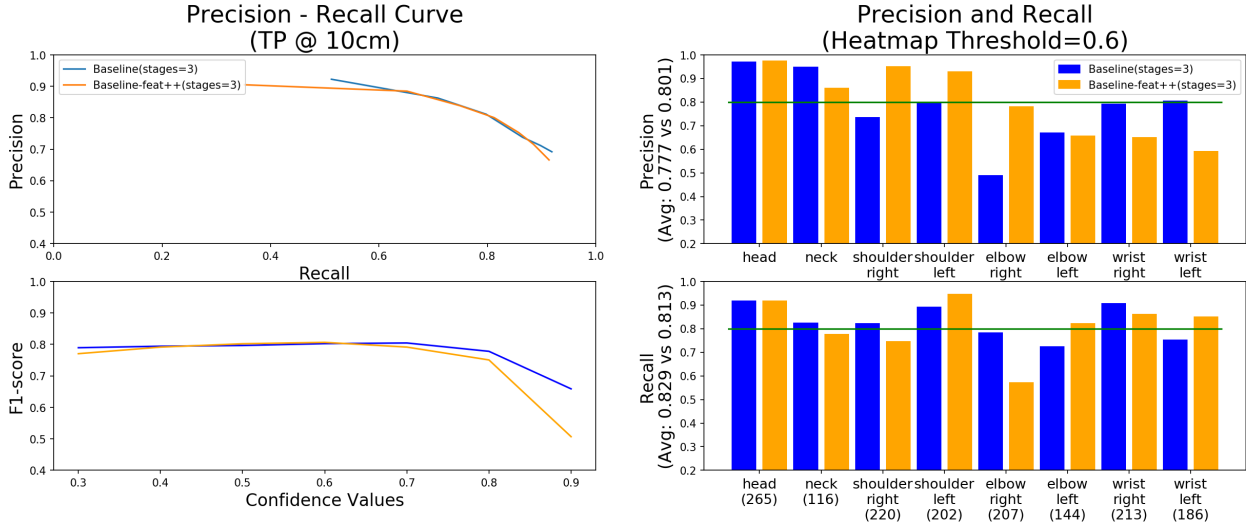


Figure 4.15: The F1-score curve's alignment with the baseline model indicates that the model shows no improvement in average performance. Specifically, the *elbow-right* body-joint has a lower recall than the baseline model.

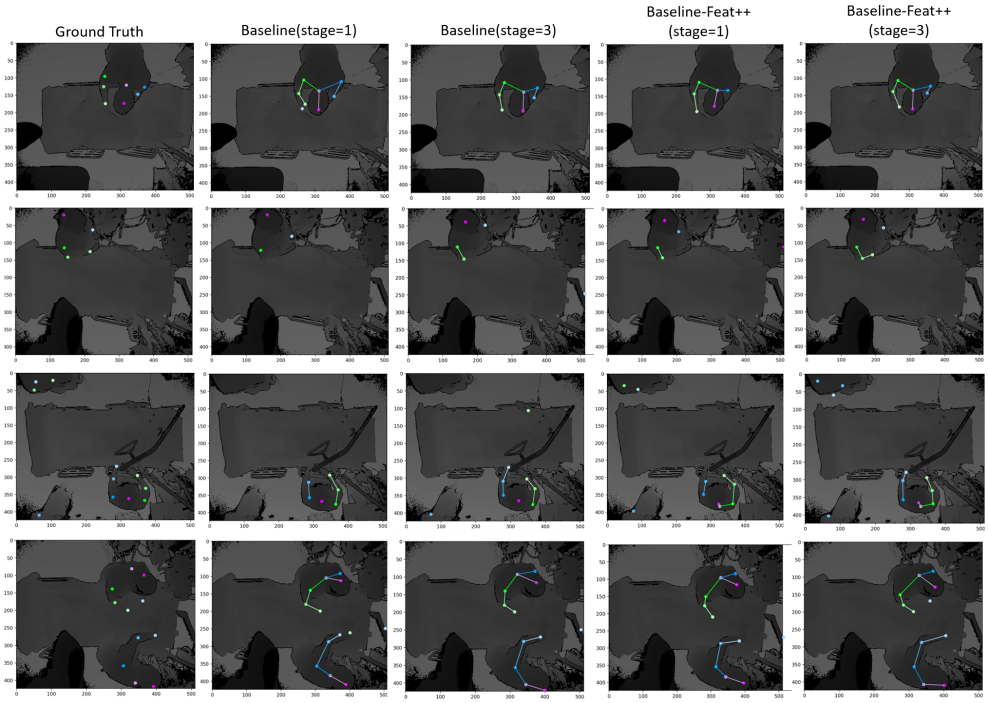


Figure 4.16: This figure compares four models to show how the *Baseline-Feat++* model makes various improvements over its baseline counterpart for the test subset of dataset 1.

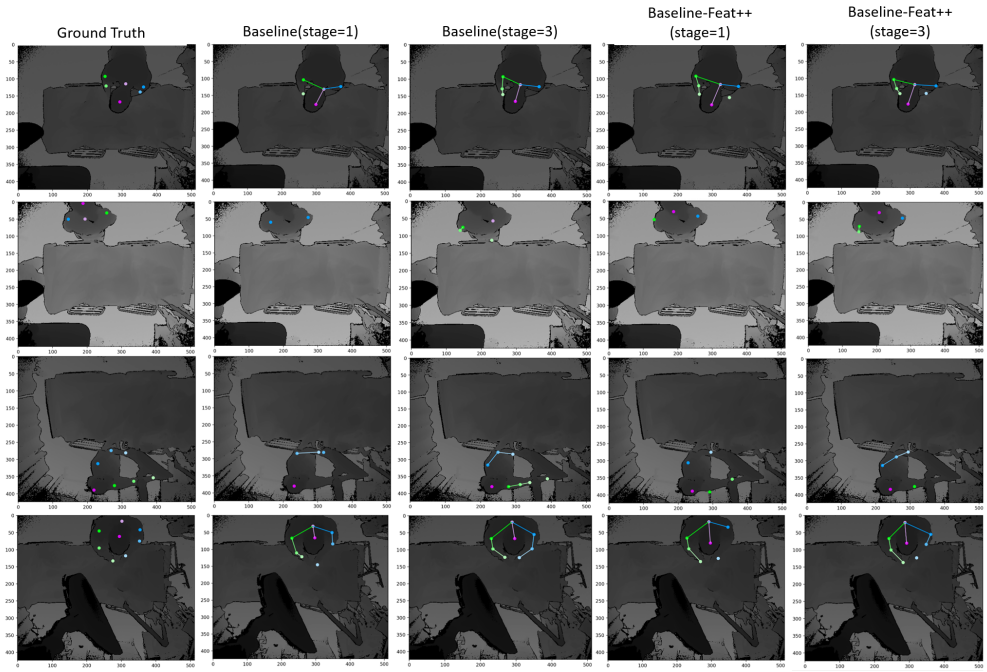


Figure 4.17: This figure compares four models to show the performance of the *Baseline-Feat++* model over its baseline counterpart for the test subset of dataset 2.

Dataset 1 The F1-score curve of this multi-stage model has shown a large improvement over the multi-stage baseline model, with the performance on low heatmap confidence values also being stable. The increase in both recall and precision has led to this models improvement over the baseline. Figure 4.16 shows examples

of multiple frames compared across the different models discussed so far. It is noticed that the increase in receptive field leads to elimination of the left-right confusion (first row), an increase in recall (second and fourth row) and a reduction in spurious detections (third row).

Dataset 2 The F1-score curves for dataset 2 do not provide a clear indication on whether additional convolutional operations in the feature representation are useful to improve the models metrics. For example, for the *elbow-right* body joint, the model replaces high precision for low recall. Figure 4.17 shows sample predictions across four models for dataset 2 to gain a visual insight into the *Baseline-feat++* models performance. While the first and second row show a success case by using the *baseline-feat++* model, the third and fourth rows show the failure cases.

4.4.2. Sequential Sub-Networks

The sequential network architecture from section 3.3 has the ideology to leverage the body part sub-network to improve the predictions of the body-joint sub-network. The model is referred to as *Baseline-Seq* henceforth, and its performance on datasets 1 and 2 can be found in figures 4.18 and 4.19 respectively.

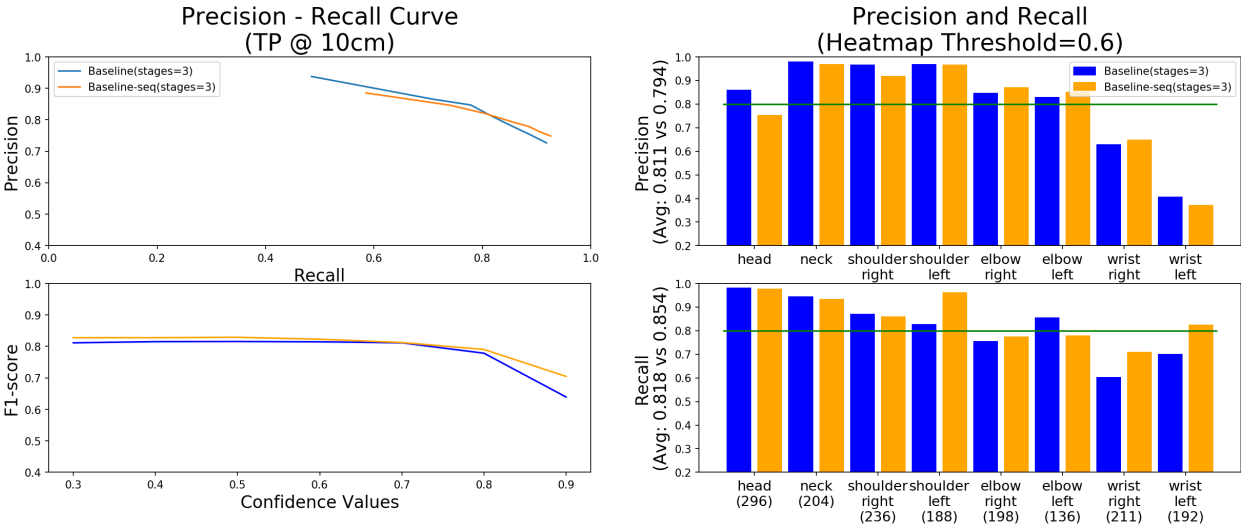


Figure 4.18: The F1-score curve of the *Baseline-Seq* model shows a very marginal improvement over the multi-stage baseline model for dataset 1. Moreover, the *wrist-left* body-joint shows no improvement in its precision.

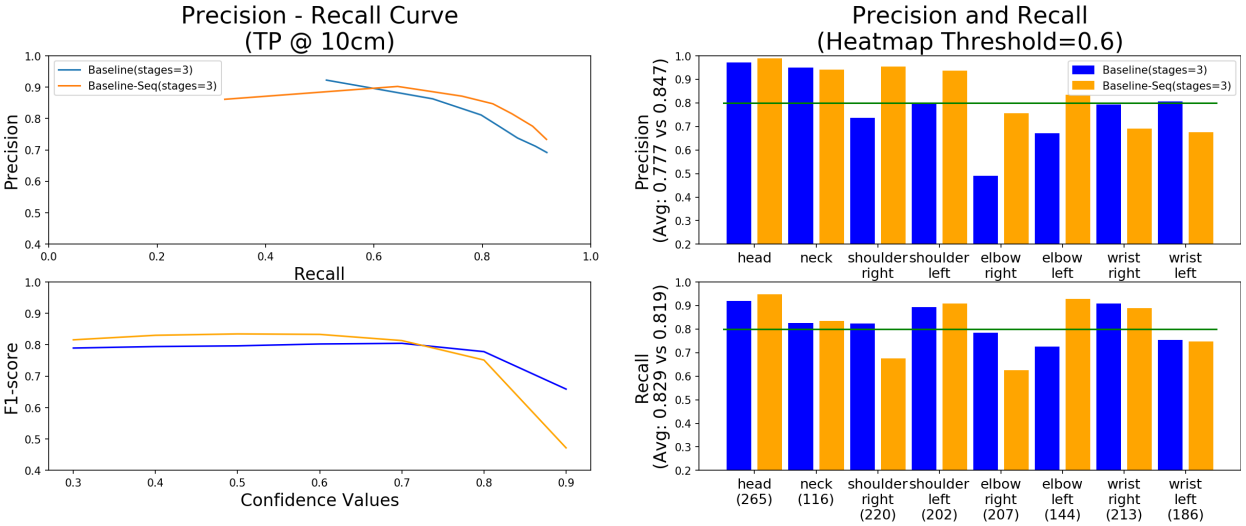


Figure 4.19: The F1-score curve of the *Baseline-Seq* model shows an improvement of a few points over the multi-stage baseline model for dataset 2. There is a drop in recall and increase in precision for the *shoulder-right* and *elbow-right* body joint.

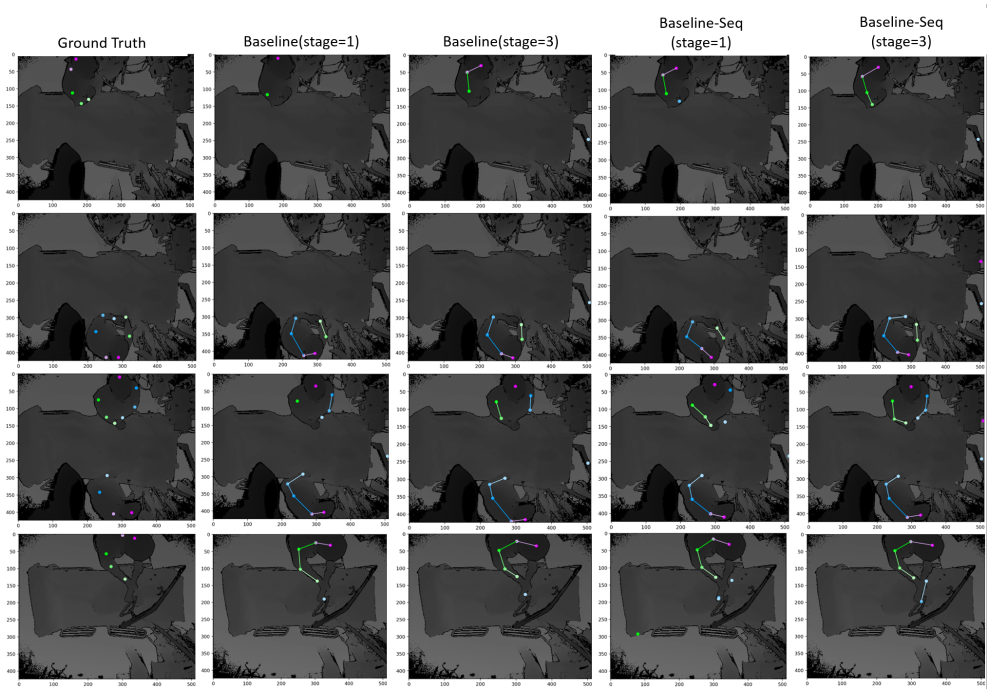


Figure 4.20: This figure compares four models to show how the *Baseline-Seq* model makes various improvements over its baseline counterpart for the test subset of dataset 1.

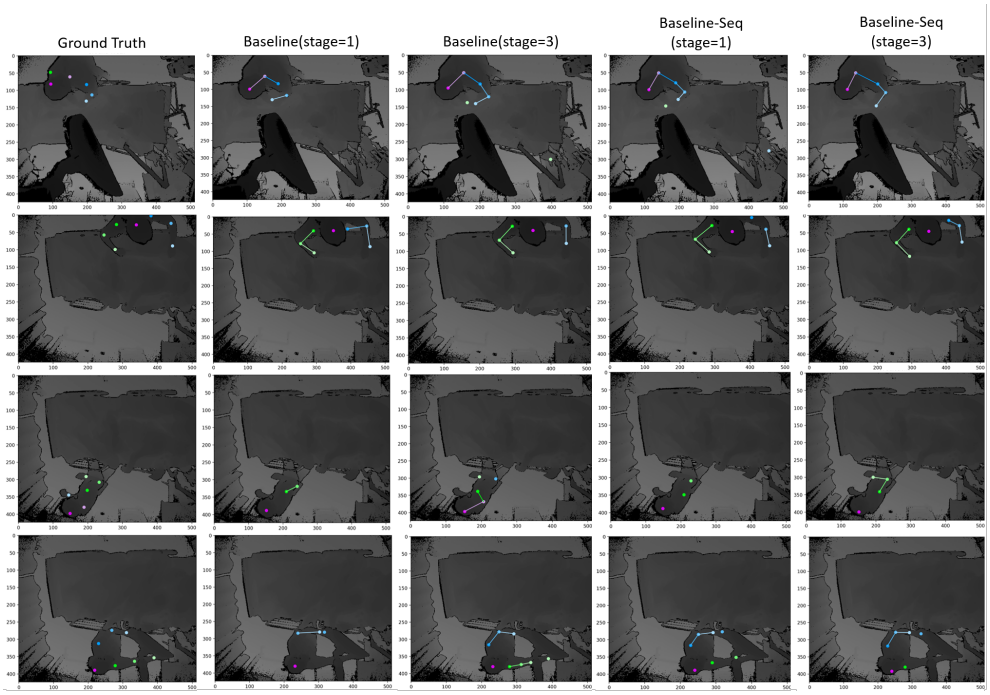


Figure 4.21: This figure compares four models to show how the *Baseline-Seq* model makes various improvements over its baseline counterpart for the test subset of dataset 2.

Dataset 1 The F1-score curve of the *Baseline-Seq* shows only a minor improvement over the multi-stage baseline model. The model only shows an improvement in average recall and minor drop in average precision while individual body-joint metrics do not show much variation. For e.g. the *wrist-left* joint still has a very low precision value due to the high number of spurious detections. Figure 4.20 shows sample predictions

with the first, second and third row showing successful examples while the the fourth row showing a failure example for the *Baseline-Seq* model.

Dataset 2 The F1-score curve for the multi-stage sequential model provides an indication for the benefits of a using a sequential model over the baseline. This model shows a decrease in recall for the *shoulder-right* and *elbow-right* body-joint while increasing the precision of both. Figure 4.21 shows a few sample predictions with the first, second and third row showing success cases and the fourth row showing a failure case for the sequential model.

4.4.3. Upsampling in Feature Extractors

The motivation behind the model described in subsection 3.3.2 is to increase the receptive field of the network by performing a single type of convolution on feature maps of different dimensions. Results for this model, referred to as *Baseline-Upsample* can be seen in dataset 1 for figure 4.22 and figure 4.23 for dataset 2.

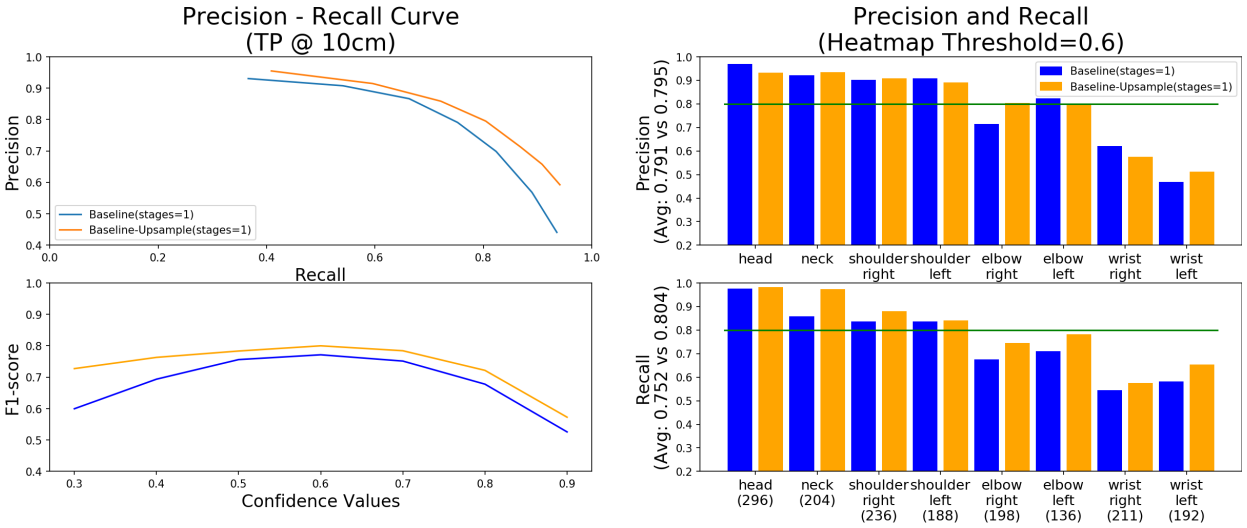


Figure 4.22: The F1-score curve of the single-stage *Baseline-Upsample* model shows an average improvement over the single-stage baseline model for dataset 1.

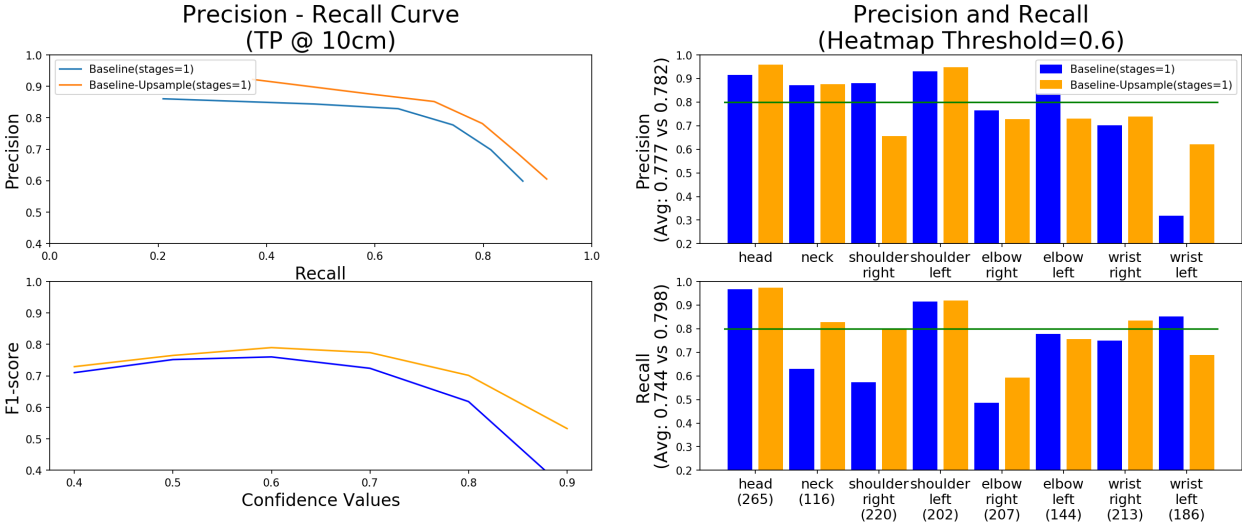


Figure 4.23: The F1-score curve of the single-stage *Baseline-Upsample* model show that there is an advantage to using this model due to an increase in the recall metric for dataset 2.

The F1-score curves show that this model holds promise since its average performance is better than the single-stage baseline. The multi-stage version of this model is not benchmarked due to a GPU-memory

overflow.

4.4.4. Proposed Models - Results Summary

Table 4.3 lists and compares the performance of the multi-stage baseline model and the proposed models for datasets 1 and 2. Performances at different TP thresholds has also been provided. Results for *Baseline-Upsample* are not shown as it is a single-stage model.

Model	**TP Threshold**	**Dataset 1**			**Dataset 2**		
		Avg Precision	**Avg Recall**	**Avg F1-score**	**Avg Precision**	**Avg Recall**	**Avg F1-score**
Baseline (stages=3)	10 cm	0.811	0.818	0.815	0.777	0.829	0.802
Baseline-Feat++ (stages=3)	10 cm	0.864	0.889	0.876	0.800	0.813	0.807
Baseline-Feat++ (stages=3)	8 cm	0.802	0.880	0.839	0.737	0.799	0.767
Baseline-Feat++ (stages=3)	6 cm	0.637	0.856	0.730	0.578	0.755	0.655
Baseline-Feat++ (stages=3)	4 cm	0.337	0.779	0.471	0.262	0.605	0.3652
Baseline-Seq (stages=3)	10 cm	0.794	0.854	0.823	0.847	0.819	0.833
Baseline-Seq (stages=3)	8 cm	0.741	0.844	0.789	0.772	0.804	0.788
Baseline-Seq (stages=3)	6 cm	0.615	0.822	0.703	0.604	0.758	0.672
Baseline-Seq (stages=3)	4 cm	0.342	0.732	0.467	0.307	0.638	0.414

Table 4.3: Human pose estimation metrics for datasets 1 and 2 on the multi-stage baseline and different proposals.

5

Discussion and Conclusion

5.1. Discussion

Human pose estimation has been a longstanding problem in the field of computer vision and hence there are myriad approaches to solving this task across various environmental settings. This work chose Convolutional Neural Nets (CNNs) over the Pictorial Structures Model (PSM) since CNNs are able to implicitly capture the relation between individual body-joints and their global configuration in a single data structure. Similar to many human pose estimation CNNs, this work adopts a heatmap regression strategy although it introduces quantization effects that are not properly addressed. A 3D CNN was chosen over a 2D CNN as the baseline for the purpose of 3D human pose estimation, as this choice allows a model to predict body-joints which are self-occluded.

In the single-stage baseline model, three classes of errors are noticed - confusion between left and right joints of the body, missing body-joints or body-parts and spurious detections on surfaces outside the bounding box scope of a non-patient stakeholder. Across both datasets, the F1-score curves for the single-stage baseline shows an F1-score below 0.75 for low heatmap confidence values. This indicates that there are patches of noisy voxels in the output heatmaps. In spite of these shortcomings the model has shown an understanding of the human body by predicting occluded or partially-occluded joints in scenes which did not possess any annotations.

The multi-stage baseline model brings with it additional computational and memory requirements, but also provides a boost to model performance. The F1-score curves of datasets 1 and 2 show an improvement over their single-stage counterparts. These curves are also stable at lower values of heatmap confidence indicating that the output body-joint heatmaps only show local peaks close to the true body-joint locations. Similar trends, albeit at higher values, also hold when the datasets are augmented with an affine transformation on the voxelized input. The results for dataset 1 are similar to those for the ITOP-top [27] dataset with the *head*, *neck* and *shoulder* body-joints performing better than the *elbow* and *wrist* body-joints.

The secondary research goal of this work was to explore techniques to incorporate additional spatial context in the baseline method. The *Baseline-Feat++* model does so by increasing the receptive field of the feature extraction module using additional convolutional operations. Another approach to increasing the receptive field of the feature extraction module would be to simply increase the size of the convolutional kernels in the feature extraction module. Although this improves the spatial spread of the convolutional operation, it does not help create a richer information hierarchy. Results on dataset 1 shown a boost in both average precision and average recall. The same level of performance gains are not seen in dataset 2 whose F1-score curves very closely align with those of the multi-stage baseline, in spite of additional parameters within the *Baseline-Feat++* model.

The *Baseline-Seq* model took a different approach on enriching the spatial context by performing body-part operations prior to body-joint operations. This was done to guide the body-joint predictions by leveraging the body-part locations. The F1-score curves of this model show an improvement over the multi-stage baseline model for both datasets. This model uses the same count of parameters as the baseline model and is able to gain a slightly better understanding of human pose with only a a difference in its data-flow logic.

The *Baseline-Upsample* experiment increases the receptive field by performing downsampling-upsampling operations within the feature extraction module. The performance boost it provides the single-stage baseline

marks it as a candidate for further investigation. It is currently unclear whether it is the use of additional convolutional operations or the downsampling-upsampling operation that leads to this improvement in performance.

To conclude, all the proposed models have shown a scope for improvement and can be used to estimate 3D human pose from depth images of indoor scenes. A combination of the *Baseline-Feat++* and *Baseline-Seq* models along with additional data augmentations may potentially boost the performance of the model. If there exists a parameter budget, use of the *Baseline-Seq* model is highly recommended.

5.2. Conclusion

The task of 3D human pose estimation is challenging due to its constituent parts possessing high degrees of freedom along with strong spatial inter-dependencies between them. This work explores the use of a top-view depth camera in an ICU room to extract human pose of multiple non-patient stakeholders in the scene. Depth cameras have been used for human pose estimation since they are privacy-preserving and also provides simple yet useful 3D information on the surfaces of a visual scene. A top-view is challenging as it leads to self-occlusion and hence a lack of visual information for those body-joints. The presence of multiple persons poses a problem on how one can assign body-joints to an individual and parse them to output a human pose.

This work proposes the use of a 3D Convolutional Neural Network (3D CNN) which takes as input a 3D point cloud extracted from the 2D depth images of the ICU scenes. A 3D point cloud contains real-world 3D information on the human body and hence the model shall not have to bother itself with any perspective distortion effects caused by the 3D-to-2D projection model. The 3D-CNN jointly learns to predict the locations of each body-joint along with the connections between them. Such a multi-task formulation allows the model to parse pairs of body-joints and assign them to each individual in the scene. It is shown that an iterative version of the 3D CNN is imperative as it allows the use of initial predictions as an informative prior.

The baseline model [77] was benchmarked using two-fold cross validation of an annotated human pose dataset. For dataset 1, the model is shown to be capable of robustly predicting the *head*, *neck* and *shoulder* joints with decreasing performance for the *elbow* and *wrist* joints. For dataset 2, the model does well on most joints except the *shoulder-right* and *elbow-right* joints. The use of the *Baseline-Feat++* model leads to overall improvement on dataset 1, but the same does not hold true for dataset 2. Thus, a hard conclusion cannot be made on the fitness of this model and further cross-validation and hyper-parameter tuning is required. On the other hand, the *Baseline-Seq* model is able to improve performance on both datasets with no additional parameters.

In theory, the main drawback of the 3D CNN approaches discussed in this work are the additional memory and computational requirements when compared to a 2D CNN. Nonetheless, the widespread adoption of GPU-based deep learning has surpassed these barriers with model inferencing in the order of milliseconds. Also, a 3D input allows the model to make predictions in 3D spaces that are not accessible using a 2D image.

Thus, this work has shown that the use of a 3D method is capable of understanding body-pose information in a scene. Furthermore, the concept of additional spatial context is a useful guiding principle to improve the robustness of existing pose estimation models.

5.2.1. Future Work

The models presented in this work attempt to find the salient regions in the visual scene i.e. non-patient stakeholders, to predict their body-joints and body-parts. Many predictions are made on 3D surfaces not belong to a human and the proposed methods do not directly address this issue. An extension of this work could be focused around directing the models attention on the regions important to the task-at-hand. The following methods could be explored for this purpose :-

- **Top-Down Approach** - A top-down approach to human pose estimation involves extracting a bounding box around a person. Using existing works on object detection [68], [42], one could send the visual content inside the predicted bounding boxes into a human pose extraction model. The assumption here is that the object detection model has to extract instances of only one class and hence shall be able to perform well. This approach allows the pose extraction framework to focus solely upon body-joint extraction. A downside to such a method is that it heavily relies on the performance of the object detection framework.

- **Multi-Task Learning** - The previous recommendation involved a two-step approach to pose extraction

with no interaction between the models during training. There is evidence in literature that performing multiple tasks using the same neural network can lead to improved supervision [47], [29]. Thus, an additional sub-network for 3D bounding box detections around persons of interest can help the network focus on the salient regions of the scene. This is a form of weak supervision since our ultimate task is not to estimate bounding boxes, bur rather specific keypoints inside them. This sub-network shall use the feature map at the end of the feature extraction module as an input. Such an approach might help prevent spurious detections on non-human surfaces of the scene.

Another approach to improve the receptive field of the network would be to employ dilated convolutions [86]. They were proposed in the context of semantic segmentation tasks for aggregating information from multiple scales, which is a requirement for human pose estimation as well. This convolutional module can directly replace the existing convolutional operations in the baseline model and provide an increase in the receptive field without any additional layers within the network.

Finally, this work only evaluates the body-joint predictions and not the body-part predictions. A metric that can evaluate properties of body-parts such as its length and accuracy of connection shall offer further insight into the performance of each model.

A

Algorithm

Evaluation Algorithm
Every annotated person in a frame has a predetermined number assigned to them which shall henceforth be referred to as the person-id (e.g. *person-1*, *person 2*). If a prediction falls inside a persons bounding box, it is assigned the person-id of that bounding box. If a prediction does not fall inside any of the boxes it is assigned a person-id of 0. The above process outputs predictions which are then parsed through algorithm 1 to obtain the values for precision and recall.

Algorithm 1: Modified calculation of TP, FP and FN

```
Data: GT([{ pid, [{'label', '3Dpos'}]} ]), PRED([{ pid, [{'label', '3Dpos'}]}])
Result: Count of TP, FP and FN
Function getMetrics()(GT, PRED, THRESH3D):
    TP = 0; FP=0; FN=0;
    for pid ∈ GT do
        for gt-obj ∈ GT[pid] do
            preds = getLabel(PRED[pid], gt-obj['label'])
            if len(preds) then
                GT_evaluated = False
                for pred-obj ∈ preds do
                    dist_pred = dist(gt-obj['3Dpos'], pred-obj['3Dpos'])
                    if dist_pred < THRESH3D and GT_evaluated is False then
                        TP = TP + 1
                        GT_evaluated = True
                    end
                    else
                        FP = FP + 1
                    end
                end
            end
            else
                FN = FN + 1
            end
        end
    end
    for pred-obj ∈ PRED[0] do
        FP = FP + 1
    end
```

Bibliography

[1] Melissa K Andrew, Susan H Freter, and Kenneth Rockwood. Incomplete functional recovery after delirium in elderly people: a prospective cohort study. *BMC geriatrics*, 5(1):5, 2005.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1014–1021. IEEE, 2009.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630. IEEE, 2010.

[4] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Discriminative appearance models for pictorial structures. *International journal of computer vision*, 99(3):259–280, 2012.

[5] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[6] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014.

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[9] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016.

[10] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017.

[11] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013.

[12] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[15] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[16] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[17] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[18] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, 100(1):67–92, 1973.

[19] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real time motion capture using a single time-of-flight camera. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 755–762. IEEE, 2010.

[20] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real-time human pose tracking from range data. In *European conference on computer vision*, pages 738–751. Springer, 2012.

[21] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real-time human pose tracking from range data. In *European conference on computer vision*, pages 738–751. Springer, 2012.

[22] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[23] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. R-cnns for pose estimation and action detection. *arXiv preprint arXiv:1406.5212*, 2014.

[24] Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear body pose estimation from depth images. In *Joint Pattern Recognition Symposium*, pages 285–292. Springer, 2005.

[25] Daniel Grest, Volker Krüger, and Reinhard Koch. Single view motion tracking by depth and silhouette information. In *Scandinavian Conference on Image Analysis*, pages 719–729. Springer, 2007.

[26] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[27] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision*, pages 160–177. Springer, 2016.

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[30] Li He, Guijin Wang, Qingmin Liao, and Jing-Hao Xue. Depth-images-based pose estimation using regression forests and graphical models. *Neurocomputing*, 164:210–219, 2015.

[31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[32] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[33] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2020, 2017.

[34] Hao Jiang and David R Martin. Global pose estimation using non-tree models. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[35] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Citeseer, 2010.

[36] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Citeseer, 2010.

[37] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[38] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013*. British Machine Vision Association, 2013.

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[40] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[42] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[43] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[44] Daniele Liciotti, Marina Paolanti, Emanuele Frontoni, Adriano Mancini, and Primo Zingaretti. Person re-identification dataset with rgb-d camera in a top-view configuration. In *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*, pages 1–11. Springer, 2016.

[45] Daniele Liciotti, Marina Paolanti, Rocco Pietrini, Emanuele Frontoni, and Primo Zingaretti. Convolutional networks for semantic heads segmentation using top-view depth data in crowded environment. In *2018 24th international conference on pattern recognition (ICPR)*, pages 1384–1389. IEEE, 2018.

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[47] Diogo C Luvizon, Hedi Tabia, and David Picard. Multi-task deep learning for real-time 3d human pose estimation and action recognition. *arXiv preprint arXiv:1912.08077*, 2019.

[48] Edward R Marcantonio, Samuel E Simon, Margaret A Bergmann, Richard N Jones, Katharine M Murphy, and John N Morris. Delirium symptoms in post-acute care: Prevalent, persistent, and associated with poor functional recovery. *Journal of the American Geriatrics Society*, 51(1):4–9, 2003.

[49] Manuel J Marin-Jimenez, Francisco J Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. 3d human pose estimation from depth maps using a deep combination of poses. *Journal of Visual Communication and Image Representation*, 55:627–639, 2018.

[50] Nancy J Martin, Michael J Stones, Janet E Young, and Michel Bédard. Development of delirium: a prospective cohort study in a community hospital. *International Psychogeriatrics*, 12(1):117–127, 2000.

[51] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Real-time convolutional networks for depth-based human pose estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 41–47. IEEE, 2018.

[52] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Real-time convolutional networks for depth-based human pose estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 41–47. IEEE, 2018.

[53] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[54] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[55] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[56] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[57] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013.

[58] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[59] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[60] Christian Plagemann, Varun Ganapathi, Daphne Koller, and Sebastian Thrun. Real-time identification and localization of body parts from depth images. In *2010 IEEE International Conference on Robotics and Automation*, pages 3108–3113. IEEE, 2010.

[61] Deva Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2007.

[62] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3433–3441, 2017.

[63] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[64] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493, 2018.

[65] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3681, 2013.

[66] Benjamin Sapp, Chris Jordan, and Ben Taskar. Adaptive pose priors for pictorial structures. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 422–429. IEEE, 2010.

[67] Benjamin Sapp, David Weiss, and Ben Taskar. Parsing human motion with stretchable models. In *CVPR 2011*, pages 1281–1288. IEEE, 2011.

[68] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[69] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011.

[70] Leonid Sigal and Michael J Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 120, 2006.

[71] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 601–608. IEEE, 2011.

[72] Min Sun, Murali Telaprolu, Honglak Lee, and Silvio Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1616–1623. IEEE, 2012.

[73] Tai-Peng Tian and Stan Sclaroff. Fast globally optimal 2d human detection with loopy graph models. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 81–88. IEEE, 2010.

[74] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

[75] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[76] Duan Tran and David Forsyth. Improved human parsing with a full relational model. In *European Conference on Computer Vision*, pages 227–240. Springer, 2010.

[77] Manolis Vasileiadis, Christos-Savvas Bouganis, and Dimitrios Tzovaras. Multi-person 3d pose estimation from 3d cloud data using 3d convolutional neural networks. *Computer Vision and Image Understanding*, 185:12–23, 2019.

[78] Michael Villamizar, Angel Martínez-González, Olivier Canévet, and Jean-Marc Odobez. Watchnet: Efficient and depth-based network for people detection in video surveillance systems. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[79] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018.

[80] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[81] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–802, 2019.

[82] LC Yan, B Yoshua, and H Geoffrey. Deep learning. *nature*, 521(7553):436–444, 2015.

[83] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, 2011.

[84] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012.

[85] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3d pose estimation from a single depth image. In *2011 International Conference on Computer Vision*, pages 731–738. IEEE, 2011.

[86] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[87] Ho Yub Jung, Soochahn Lee, Yong Seok Heo, and Il Dong Yun. Random tree walk toward instantaneous 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2015.

[88] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[89] Youding Zhu, Behzad Dariush, and Kikuo Fujimura. Controlled human pose estimation from depth image streams. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[90] Silvia Zuffi, Oren Freifeld, and Michael J Black. From pictorial structures to deformable structures. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3546–3553. IEEE, 2012.