Institute of Automatic Control

RWTH AACHEN UNIVERSITY

The present work was submitted to the Institute of Automatic Control of RWTH Aachen University

Utilizing and Adapting Synthetic Data for Pose Estimation of Occluded Humans in Horizontal Positions

Master's thesis
of
Atharva Suhas Vaidya, B.Eng
Matriculation Number 416254

Field of study:	Robotic Systems Engineering
Duration of this work:	18 Weeks
Submitted on:	08.01.2024
Number:	S2271
Supervisor:	Tim Redick, M. Sc
External supervisor:	Dr. Thomas Lindemeier

Ich versichere, diese Arbeit im Rahmen der am Institut üblichen Betreuung selbständig angefertigt und keine anderen als die angegebenen Quellen verwendet zu haben.

(Atharva Suhas Vaidya)

Aachen, 08.01.2024

Contents

List of Figures

List of Tables

1 Motivation and Task Description

Computer Vision (CV) is a rapidly evolving field that deals with interpreting images, videos, and other sensory data. It is a core component of many artificial intelligence applications, including image recognition, object detection and tracking, and Human Pose Estimation (HPE). HPE is the extraction of body configurations in images or videos. Typically, it is the inference of joint coordinates and reconstruction of a human skeletal representation [18]. HPE enables machines to perceive and interpret human body postures and movements. This has found numerous applications in human-machine interaction, activity recognition and monitoring, surveillance and security, animation, and augmented reality. However, the accurate detection of human poses is often challenging particularly because of occlusions, different point-of-views, and the lack of accurate ground truth data. This thesis deals with an important aspect of HPE that has been relatively left unaddressed - pose estimation of humans in horizontal, lying positions with severe occlusions using RGB images. The utilization of synthetic datasets in the training process is explored to overcome the under-representation of such poses and occlusions in other HPE datasets.

1.1 Motivation

While HPE has been a topic of significant research interest, existing models primarily perform well on humans in upright, unobstructed positions, such as walking and standing in day-to-day scenarios. In real-world applications, especially in healthcare and security applications, humans are frequently in horizontal positions, such as during sleep or medical procedures, and frequently obstructed by objects like blankets. Heavy occlusions can significantly reduce the amount of visual information, making it difficult for pose estimation algorithms to accurately localize and identify keypoints as seen in Fig. 1.1. Similarly, pose estimation accuracy can decline significantly in scenarios with extreme camera perspectives, such as those from 360-degree cameras or top- or bottom-mounted setups, as well as in situations involving extreme human poses, such as during sleep or in healthcare and sporting events. It captures human poses that deviate from the norm observed in conventional datasets or everyday scenarios. These unique camera angles and poses introduce a degree of complexity not typically encountered in standard pose estimation challenges.

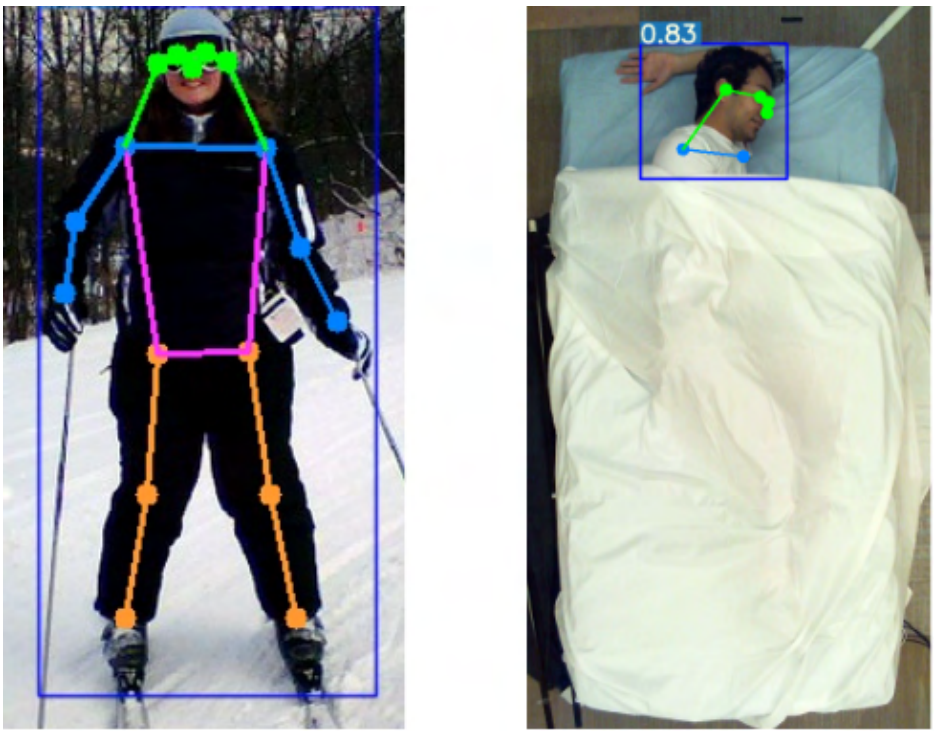


Figure 1.1: Inference result of human pose estimation shown in a conventional situation (left) and a horizontal position with heavy occlusion (right).

This research is motivated by the practical significance and benefits of accurately estimating the human pose in these horizontal and occluded conditions. In healthcare, such research can lead to more accurate patient monitoring, particularly in critical care units and surgical rooms where patients are frequently lying and covered with surgical drapes or other equipment. In the field of security and surveillance, the ability to estimate poses in occluded and horizontal scenarios enhances the detection and tracking of persons in non-standard conditions. Moreover, this research has potential applications in collaborative robotics and human-machine interaction, improving the ability of machines to understand and respond to human postures in varied environments with heavy occlusions. Furthermore, many existing methods in this domain try to solve this problem by using many different input modalities requiring expensive hardware. Hence, to eliminate the reliance on such sophisticated equipment, this study focuses on using simple RGB images and fast detectors.

Synthetic data has emerged as a valuable tool in various fields, particularly CV and machine learning. Specifically, synthetic data can address the limitations of real-world datasets, which often lack sufficient examples of rare events, edge cases, or challenging scenarios specific to a certain application. Focusing on the problem of pose estimation under occlusion, synthetic data offers a promising solution. A vast array of occluded poses and scenarios can be generated, providing pose estimation algorithms with more diverse and challenging training examples. This can improve robustness and generalization capabilities, particularly for occluded poses. The research

will focus on developing methods to effectively adapt and enhance synthetic data to effectively train deep learning models and hence, boost their performance in real-world applications.

Additionally, the thesis will explore techniques for incorporating synthetic data into training pose estimation algorithms to enhance their performance under occlusion. The research findings can contribute to the broader field of CV by providing insights into the effective use of synthetic data for improving the robustness and generalization of the existing pose estimation algorithms or to apply and re-purpose these methods for a specific use case. The developed methods and techniques can be extended to other pose estimation tasks and potentially applied to other CV problems.

1.2 Task Description

This section briefly describes the previous work that forms the basis of this thesis. It is followed by an overview of the objective of the thesis and the tasks carried out to achieve the objective as well as the scale used to evaluate the results obtained. Lastly, a short outline is provided that looks into the upcoming chapters of this thesis.

1.2.1 Generation of Synthetic Data

The dataset creation process was conducted during my internship at Carl Zeiss AG, during which the synthetic representations of human poses in horizontal, sleeping positions were developed and refined. As a result, an initial version of the synthetic dataset was available at the commencement of this research. The synthetic data utilized in this research was generated using Blender[1], a 3D modeling and rendering software widely used for synthetic data generation. It provides a wide range of tools and features that enable the modeling of realistic and diverse 3D objects and environments, including human models, clothing, and other complex objects. It also involves advanced physics simulation with the Bullet Physics engine. With the help of Blender, a diverse, realistic, comprehensive, and highly customizable dataset for training a Convolutional Neural Network (CNN) can be generated. This approach can be cost-effective, scalable, and more accurate than the traditional data collection methods for pose estimation.

For the generation of this dataset, more than 40 different character models were available for use in the company. The characters were rigged with a skeleton (or armature) containing more than 100 bones resulting in a very detailed and customizable armature. The characters also had realistic and diverse clothing and accessories. They represented a varied range of humans in terms of age, gender, body shape, height, skin color, etc. (see Fig. 1.2). During the previous work, the ways to manipulate the characters in a particular Blender scene using its Python API, for example, changing their poses, changing the material parameters of the clothing and skin, etc. were developed. Also, a Docker-based pipeline for simplifying the process of character and scene manipulation, image rendering, and generating annotations was designed.

[1] Blender 3.5.1: an open source 3D creation suite available online at `https://download.blender.org/release/`

Figure 1.2: Diversity of character models

1.2.2 Image Modality and Adaptation of Synthetic Data

Focusing on localization and pose estimation of humans in horizontal, sleeping positions using only RGB images, this thesis aims to develop a robust system for pose estimation under severe occlusion caused by blankets and other objects. By incorporating synthetic data into the training pipeline of pose estimation models, the dependence on real datasets is aimed to be reduced. This thesis also focuses on the adaptation and diversification of the synthetic dataset created in the previous work to improve its effectiveness in training the models. The use of RGB images in this research is advantageous as it enhances accessibility and affordability, reducing the dependency on expensive Infrared (IR) and depth cameras.

1.2.3 Training and Metrics

The quality of the synthetic data generated will be checked indirectly by evaluating the real-world performance of HPE models trained on synthetic data. The thesis will investigate the State-of-the-Art (SOTA) models and explore techniques such as Transfer Learning (TL) for training these models with synthetic data. By employing well-established architectures, the objective is to improve the robustness of localization and pose estimation even under severe occlusion. Achieving a high number of true positives in such conditions is essential, hence the target is to enhance the precision and recall-based pose estimation metrics of the network. The effect of including a small amount of real data in the training process is also explored. The goal is

to reduce the reliance on real data for achieving good performance metrics, and this objective is pursued through the enhancement of synthetic data quality. Later, the potential of Domain Adaptation (DA) techniques to boost accuracy by using unlabeled real data in the training process is also examined.

1.3 Outline

First, a thorough literature review of CV algorithms for HPE, the evaluation metrics used in such algorithms, and the use of synthetic data in CV algorithms are presented. The insights gained, guide the selection of the models for the subsequent training process. Additionally, the literature review explores the utilization of synthetic datasets in CV applications. This provides answers to critical questions about the qualities and characteristics of an effective synthetic dataset.

Furthermore, the specific modifications chosen for the synthetic dataset are explained along with their intended effects on the performance of the trained models. Subsequently, the selected models are described especially the architectures and the modifications made. The real counterpart to the synthetic dataset chosen for the evaluation of the models is then described. The experiment results by training the models with the generated synthetic data are then described and analyzed. Based on the best-performing model and configuration, further comparisons are made by including varying amounts of real data. Real datasets introduce additional complexities and challenges not fully captured in the synthetic dataset. This will give insight into the effectiveness of the synthetic data, the existing domain gap between the datasets, and the amount of real data that is still necessary for a good performance. Finally, a DA method is integrated into the training of a HPE model with the synthetic datasets, and the results are examined.

2 Fundamentals of Vision-Based Human Pose Estimation

To highlight the contributions of this thesis to the field of CV, a short history of the field including the first works, the introduction of machine learning methods, and the latest developments to tackle more complex problems will be explained. Then, a short introduction to deep learning will be provided. By now, deep learning has become a widely accepted standard for a majority of tasks in CV due to its superior performance compared to conventional methods. Hence, deep learning-based pose estimation methods are extensively used in this thesis. A brief introduction of the core concepts in deep learning like feed-forward neural networks and CNNs is given.

After this general introduction, the relevant tasks for this thesis like object detection and pose estimation will be introduced. Furthermore, the common datasets and metrics necessary to evaluate the methods used in this thesis along with the SOTA methods will be described. The last section will provide a brief review of the use, benefits, and limitations of synthetic data in CV applications.

2.1 History of Computer Vision

Lawrence Roberts is widely considered the father of CV. His PhD thesis published in 1963 titled "Machine perception of three-dimensional solids" [71] was the first work on CV based on machine learning techniques. It laid the foundation for future research on learning through performing matrix manipulation. Later in 1980, Kunihiko Fukushima published a paper [23] which introduced an early form of a neural network with multiple connected layers, called *Neocognitron*. It was able to recognize basic visual patterns anywhere in the image. In the 90s, machine learning algorithms such as Support Vector Machines (SVM) along with Artificial Neural Networks (ANNs) were developed and gained popularity in tackling more complex problems like image classification.

In the 2000s, development in CNNs saw huge strides due to advances in computational power. One of the earliest CNN-based image classification model, LeNet-5 was published in 1998 which contained a series of convolutional and subsampling layers followed by full-connected layers (see Fig. 2.1) [41]. Later in 2012, an improved classification architecture, AlexNet [37] was introduced in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [74]. It achieved an error 10.8% points lower than the second-best model on the ImageNet dataset consisting of 1.2 million images containing over 1000 object categories. This was largely possible due to training on Graphics Processing Unit (GPU). It was a large model for the time that contained 60 million parameters. With the emergence of architectures like VGG, GoogleNet, and ResNet in the 2010s,

it became evident that the increase in depth of the CNN directly increased the corresponding accuracy [80, 81, 33]. This proved the power of deep CNN architectures which paved the way for applications in other areas of CV.

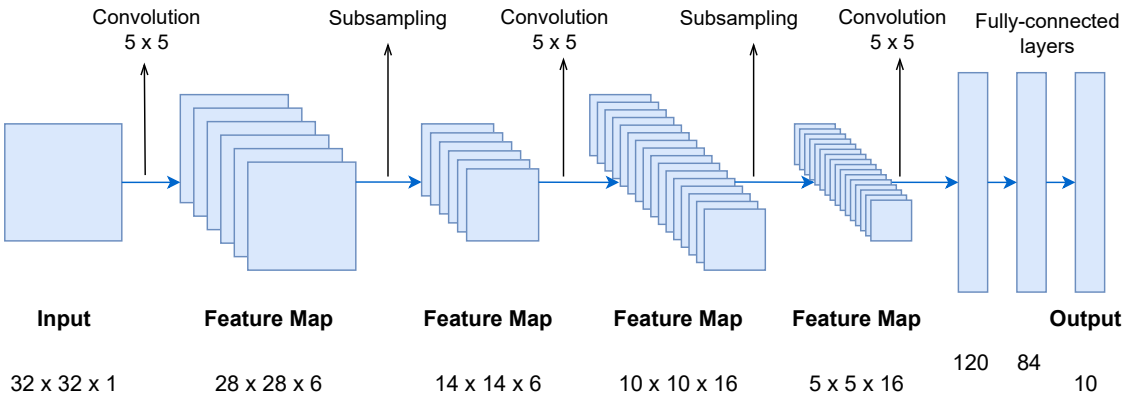


Figure 2.1: LeNet-5 model architecture [41]

2.1.1 Feed-forward Neural Networks

A feed-forward neural network is the simplest form of an ANN. Information may pass through multiple hidden nodes but is only processed in one direction without any loops, hence the name. The simplest form of a feed-forward neural network is a single-layer perceptron. Here, the inputs enter the layer and are multiplied by a unique weight. These values are then added together to get a weighted sum of the series of inputs. The sum is then passed through an activation function that usually sets the output to 1 if the sum is above a certain threshold and 0 or -1 otherwise. A series of multiple layers of interconnected perceptrons form a feed-forward neural network. These layers typically include an input layer, one or more hidden layers, and an output layer as seen in Fig. 2.2. Common activation functions include the sigmoid, ReLU, and tanh. The training of these networks takes place with a process called back-propagation. The weights of each node are adjusted such that the error between its predictions and the actual target values is reduced.

2.1.2 Convolutional Neural Networks (CNNs)

A CNN is a specialized type of ANN designed to effectively process visual data, such as images and videos. CNNs have been extremely successful in CV applications such as object detection and tracking, face recognition, pose estimation, etc. due to their ability to handle extremely high-dimensional input. A CNN usually consists of three types of layers: convolutional layers, pooling layers, and fully connected layers. In convolutional layers, various kernels are used to convolve over the whole image (or the intermediate feature maps of the image). Each node in a convolutional layer receives input from a cluster of neighboring nodes in the previous layer. Informally, shallower layers in an ANN are the layers closer to the input layer, while deeper layers are layers distant from the input. The neurons in the shallow layers are capable of extracting

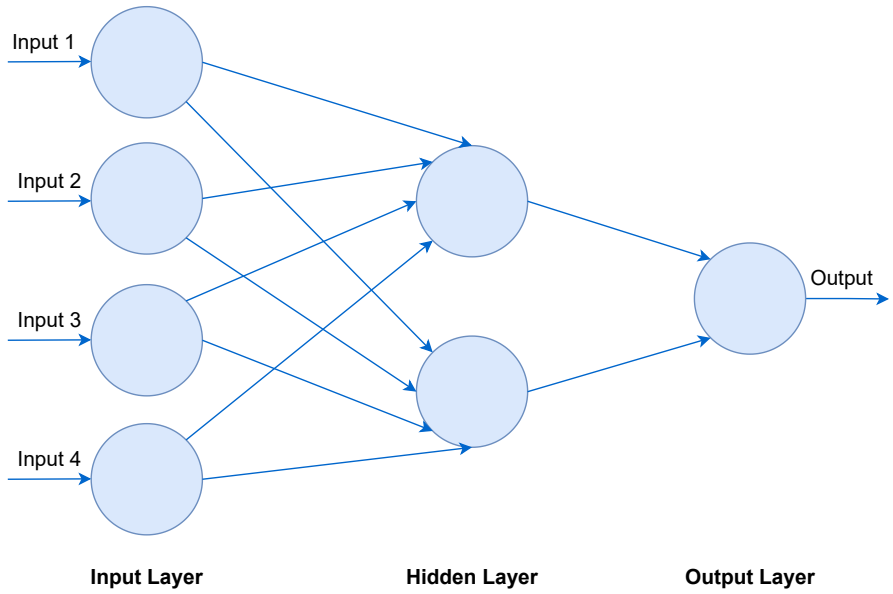


Figure 2.2: A simple three-layered feed-forward neural network

elementary features such as lines and edges from the image. Higher-order features can then be detected in deeper layers based on such elementary features. The nodes of a convolutional layer are organized in multiple planes and all nodes in a plane share the same weights. Thus, each such plane constructs a specific feature from the image and outputs the feature map. The pooling layers are tasked with reducing the spatial dimensions of the resulting feature map from the convolutional layers. It does not affect the depth dimension. This results in a loss of information, which is beneficial for reducing computational load as well as avoiding overfitting. Fully connected layers are usually at the end of CNNs and are responsible for high-level reasoning depending on the task. These layers convert 2D feature maps into 1D feature vectors which can be further processed [88].

A large number of parameters need to be learned for a CNN which also needs a large amount of training data. A high number of parameters can also hinder generalization and may lead to over-fitting. Many techniques such as data augmentation, dropout, pre-training, and stochastic pooling are utilized to overcome this problem. One of the biggest advantages of CNN is its ability to deal with variance in scale and translation, making it especially useful in tasks such as object detection [88].

2.2 Object Detection

Object detection is the task of accurately detecting instances of objects in an image or a video and predicting the bounding box around it. It is often a preliminary step in many CV tasks such as object tracking, image segmentation, face recognition, and pose estimation.

2.2.1 Traditional Methods

Most of the traditional methods of object detection before deep learning were based on hand-crafted features. In 2001, Viola Jones detectors [86, 87] were able to achieve real-time object and human face detection. They used sliding windows that covered all possible locations and scales in an image to search for a human face. Later, Dalal *et al.* introduced Histogram of Oriented Gradients (HOG) feature descriptors which as motivated primarily towards pedestrian detection [15]. To detect objects at every scale, it kept the size of the window constant and instead re-scaled the input image multiple times. Deformable Part-Based Model (DPM) [22] was an improvement over HOG. DPM-based methods won three PASCAL VOC challenges [20] and are considered the epitome of traditional object detection methods.

2.2.2 CNN-Based Two-Staged Detectors

Later methods in object detection benefited vastly from the advancements of CNNs. The first of such methods was Regions with CNN features (RCNN) [29, 30]. There are two main families of methods in the deep learning era of object detection: two-staged detectors and single-staged detectors. RCNN is a two-staged detector that first extracts object proposals in the image which is then fed to a pre-trained CNN model for feature extraction. Lastly, classifiers like SVM are used to predict and categorize the object in the region. Many further improvements were proposed to develop SPPNet [32], Fast RCNN [28], Faster RCNN [69], and others. Faster RCNN [69] utilize Region Proposal Network (RPN), a CNN that generates rectangular object proposals or regions of interest from an input image. Later, non-maximum suppression is applied to the proposal regions. And finally, Fast RCNN is used to generate object proposals. Two-stage detectors can achieve very high accuracy but are rarely employed in engineering due to their lower speed and higher complexity [98].

2.2.3 CNN-based Single-Staged Detectors

In recent times, single-stage detectors have become hugely popular thanks to their ability to detect all objects in a single inference step. You Only Look Once (YOLO) [66] was the first single-stage detector. It achieved better than real-time speed on the PASCAL VOC challenge. YOLO divides the entire image into regions and predicts bounding boxes and probabilities for each region simultaneously. The bounding boxes can then be ranked simply by the overall probability. This style of detector can be trained end-to-end with a single loss function. The losses for incorrect predictions of the boxes, incorrect locations, and incorrect classes can all be combined into a single loss. It could achieve results close to the best Faster RCNN in real-time. YOLO's performance suffers noticeably when detecting dense and small objects [98]. Due to its speed, practicality, and easy-deployed features, it gained a lot of popularity. It gave rise to many different versions of YOLO [67, 68, 5, 26].

Later in 2016, Liu *et al.* [52] proposed a Single-shot Detector (SSD). SSD uses predefined anchor boxes for prediction. Anchors are a set of pre-defined plausible bounding boxes for the classes

we want to detect that are used during training and inference [69]. It applied this idea on multiple scales to detect boxes of various sizes. SSD performed even faster than the original YOLO with performance on par with the Faster RCNN. Another notable single staged detector is the RetinaNet [45] which introduced a new loss function called focal loss. It addresses a problem in normal single-staged detectors, that even the correctly classified examples contribute to the classification loss. So it used an additional polynomial factor $-(1-p)^{\gamma}$ for some $\gamma > 0$ in the normal cross-entropy loss $-\log p$ which reduces the loss of a correctly classified example to close to zero. It has since, been used in many deep-learning models. Very recently, YOLOv7 [89] developed by the YOLOv4 team has been introduced. It outperforms most existing object detectors in terms of speed and accuracy [89]. They also propose a HPE method by integrating YOLOv7 with YOLO-Pose [55].

Just a few months later, Ultralytics[1] proposed YOLOv8[2], a follow-up on their YOLOv5[3]. Although these models are from the YOLO family, their architectures are quite different. YOLOv7 utilizes the extended efficient layer aggregation networks as the backbone. They also introduce coarse-to-fine lead-guided label assignment to solve the problem of assigning dynamic targets for the outputs of different branches. They also propose various strategies like planned re-parameterization of the model to increase robustness, and extend and compound scaling methods for increased computing efficiency. The backbone of YOLOv8 is based on the CSPDarknet53 [90]. It adopts an anchor-free model head to improve accuracy and efficiency. Anchor-free methods predict directly the center of an object instead of the offset from a known anchor box. YOLOv8 also utilizes specific training routines like mosaic augmentation to boost accuracy. They provide multiple different models varying in size and complexity and therefore accuracy and speed. Apart from object detection, they have also provided models for pose estimation, image classification, and segmentation. The variety of CV tasks tackled by the family of YOLO algorithms make a good case for their versatility owing to their simple architecture and their efficient and powerful feature extractors.

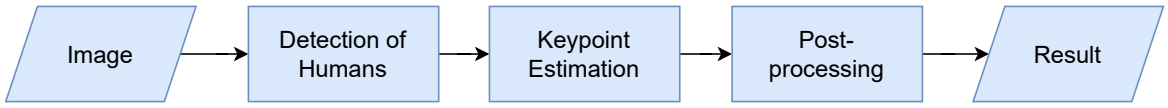
2.3 Human Pose Estimation (HPE)

HPE models face many challenges, of which the variance in RGB data due to lighting conditions, occlusions, and variance in appearance, clothing, textures, and backgrounds is a major one. Another challenge is the need for a large amount of training data to effectively learn the complex poses and relationships between the joints of the human body. HPE is usually a very specific problem depending on many factors that govern the approach one needs to take to solve it accurately. Hence, finding one SOTA model that can be effective out of the box in every use case is challenging.

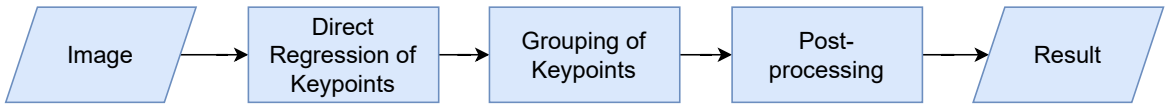
[1] Ultralytics docs: `https://docs.ultralytics.com/`
[2] YOLOv8 GitHub: `https://github.com/ultralytics/ultralytics`
[3] YOLOv5 GitHub: `https://github.com/ultralytics/yolov5`

2.3.1 Top-down and Bottom-up Approach

There are two main methodologies for HPE: Top-down and bottom-up approach. The top-down approach detects and localizes the humans in the image or video first and then estimates the body joints within the detected bounding box (Fig. 2.3a). It is then followed by calculating the pose. The bottom-up approach directly estimates the location of the human body parts in the image (Fig. 2.3b). It is then followed by grouping them to the humans they belong to and then finally calculating a unique pose [16].



(a) Top-down methods



(b) Bottom-up methods

Figure 2.3: Human pose estimation frameworks

In recent years, both methodologies have been extensively explored using deep learning methods. Both approaches have their inherent advantages and disadvantages. Many top-down approaches are simply the combination of existing SOTA object detectors and a pose estimator head [61]. The top-down approaches usually suffer from the problem of early commitment to the detection of humans. It means that the entire process of keypoint estimation depends on the accuracy of the human detectors. If the human is not detected in the first stage, there is no possibility of recovery and hence, there will be no keypoints detected for that human [59]. The accuracy of the detected keypoints also depends vastly on the accuracy of the bounding box of the human. This also means that if the predicted bounding box does not include the entire person, the joints lying outside the box have essentially no chance of getting predicted. Top-down methods also suffer from time complexity as the computational cost scales linearly with the number of people in the frame. It means that for every person in the frame, a single-person pose estimator is run [59]. However, in recent times, many winners of pose estimation challenges are top-down approaches and even the best performing bottom-up methods fail to reach the accuracy of the SOTA top-down approaches [6, 9, 60]. Many of the recent surveys also show that the top-down approaches consistently outperform the bottom-up approaches [8, 40]. Furthermore, the performance of these methods can be boosted with improvements in the human detectors as well as single-person pose estimators [40]. Both of these areas have enjoyed significant research interest in recent times.

Bottom-up approaches do better with time complexity as they eliminate the need for object

detectors and directly perform keypoint estimation. This in turn gives rise to a new problem: How to judge the identities of estimated joints [8]? This problem is aggravated when the keypoint is under occlusion or self-occlusion. Hence, the performance in such cases might be worse than the top-down methods depending on the grouping method applied. The time required for grouping is usually not of huge concern if the method is designed properly. This approach also suffers from scale variability of humans in the image as the network fails to learn scale-invariant and consistent features from the images [16]. Hence, it often relies on large input resolution and multi-scale training to reach SOTA accuracy [60].

2.3.2 Evaluation Metrics

To compare the accuracy of different HPE methods, many different evaluation metrics exist across multiple benchmark datasets. The designers of the dataset often recommend different metrics. Some of the most common metrics are explained below:

- Mean Per Joint Position Error (MPJPE) is one of the most common metrics used mainly for 3D HPE. It is often referred to as mean reconstruction error or 3D error. MPJPE is the mean of the Euclidean distances between the ground truth coordinates and the estimated coordinates for each joint. It is a good baseline metric that can be used to estimate a wide variety of methods.

- Mean Per Joint Velocity Error (MPJVE) is used when a pose sequence is extracted from a video. MPJVE is the mean per joint of the first derivative of the 3D pose sequences. It is used to measure the smoothness of predictions over time.

- Thresholds Metrics is a family of metrics that involve defining a threshold where a joint position is correctly detected. Percentage of Correct Parts (PCP) sets the threshold at half the size of the ground truth joint segment to determine if a prediction over a limb segment is correct. It can also be modified for 3D HPE. The issue with this metric is that a shorter limb will be less likely to be considered detected as the threshold decreases. Percentage of Correct Keypoints (PCK) is another threshold metric for 2D HPE. It uses a subject-specific threshold for each joint instead of limbs. This metric does not have the issue with shorter limbs as it is self-adapting to subjects with different proportions.

- Object Keypoint Similarity (OKS) was first introduced in COCO competition as a variable to compute the Mean Average Precision (mAP). For understanding the OKS metric, Intersection over Union (IoU) also needs to be explained. IoU is a very common metric in object detection. OKS is treated as IoU for keypoint estimation tasks. IoU is defined as the ratio between the area of intersection to the area of union (see Fig. 2.4). It helps estimate the accuracy of the bounding box prediction. An IoU of 1 indicates perfect overlap of the predicted and ground truth bounding boxes, whereas 0 indicates no overlap.

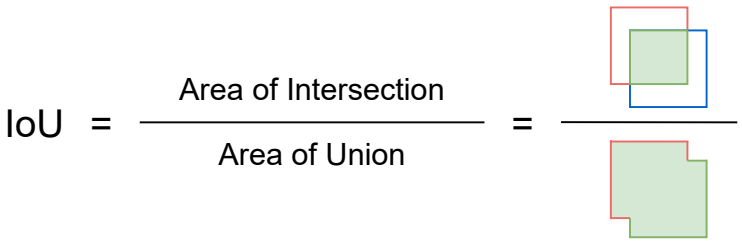


Figure 2.4: Intersection over Union (IoU)

Unlike object detection, which needs an area-based metric, for keypoint estimation, a distance-based metric is needed. Hence, COCO introduced OKS. It is calculated from the distance between the predicted joints and the ground truth points normalized by the size of the human. The scale and keypoint constants need to equalize the importance of each keypoint. It can be calculated as follows[4]:

$$OKS = \frac{\sum_i \left[\exp\left(-\frac{d_i^2}{2s^2k_i^2}\right)\delta(v_i > 0)\right]}{\sum_i \delta(v_i > 0)}$$

where i is a joint index, d_i are the Euclidean distances between the predicted keypoint and ground truth, s is object scale and k_i are the keypoint constants that control falloff given by the COCO dataset, $\delta = 1$ if $v_i > 0$ and 0 otherwise, and v_i is the visibility flag of the ground truth

$$v_i = \begin{cases} 0 & \text{if not labeled} \\ 1 & \text{if labeled but not visible} \\ 2 & \text{if labeled and visible} \end{cases}$$

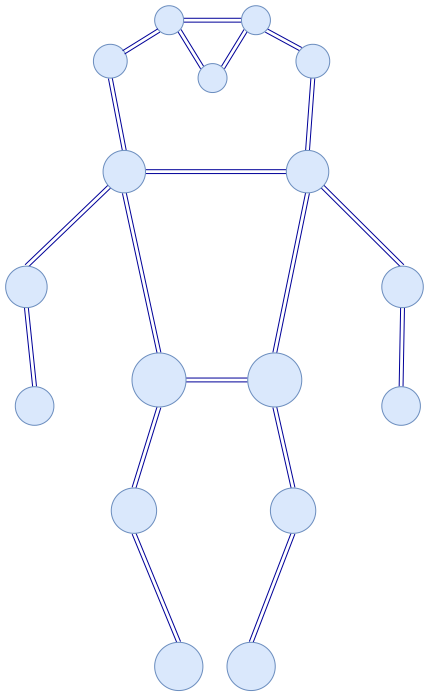
For each keypoint, a similarity index ranging between 0 and 1 is obtained. These similarity indices are averaged over all the labeled keypoints (keypoints for which $v_i > 0$). Predicted keypoints that are not labeled ($v_i = 0$) do not affect the OKS. Perfect predictions for all the keypoints of a person will yield OKS = 1. When the predictions for every keypoint are far from the ground truth by more than a few standard deviations will have OKS $\approx$ 0. The per-keypoint standard deviation σ_i with respect to the object scale was measured by COCO using the 5000 redundantly annotated images in the validation data. These standard deviations (σ_i) are shown in Fig. 2.5 for all the keypoints. The size of the circle roughly denotes the standard deviation range for each keypoint. The per-keypoint constant k_i is set to twice of σ_i.

[4] See `https://cocodataset.org/#keypoints-eval`

The number of True Positives, False Positives, and False Negatives are decided based on the OKS threshold. All unmatched keypoints are considered to be False Negatives. In a similar manner as in object detection, the average precision can then be calculated as

$$AP = \frac{TP_{(OKS>td)}}{TP_{(OKS>td)} + FP_{(OKS\leq td)}}$$

where TP and FP are the numbers of true positive and false positive, respectively, and td is a OKS threshold. The mAP_{50} is the AP at $td = 0.5$ and $mAP_{(50-95)}$ is the mean AP over ten OKS thresholds at $(0.50, 0.55, ..., 0.90, 0.95)$. These are common metrics to evaluate 2D multi-person HPE for many of the recent datasets. mAP_{50} is considered to be a lenient metric whereas mAP_{50-95} is considered to be a strict metric because of the weight it gives to the higher OKS thresholds. Most of the models use the mAP_{50} as default.



Keypoint	σ_i
Eyes	0.025
Nose	0.026
Ears	0.035
Shoulders	0.079
Elbows	0.072
Wrists	0.062
Hips	0.107
Knees	0.087
Ankles	0.089

Figure 2.5: Per-keypoint standard deviations (σ_i) in OKS loss across 17 human keypoints. The size of the circle roughly represents the area of standard deviation for each keypoint.

2.3.3 Recent Methods

Recently, ANN-based methods made good improvements in the accuracy of HPE. Even some generic neural networks with efficient architectures show great potential. Newly emerging Transformer-based models [92, 57, 93, 43] are proving to be a potential alternative to CNNs. ViTPose is the recent SOTA method that uses a simple non-hierarchical vision transformer as a backbone to generate a feature map [92]. It first embeds the image into tokens with a patch

embedding layer. These tokens are passed through several transformer layers. This makes up the encoder part of ViTPose. A lightweight decoder is then used to process these features. It consists of deconvolution layers and one prediction layer which perform upsampling of the feature maps and keypoint regression respectively. It shows good performance on COCO dataset [46] (see Fig. 2.6), however with a much larger model. Many challenges are still unanswered, especially regarding generalization and robustness. Compared to CNNs, transformers usually depend on larger datasets for training. Their model architectures are typically larger than CNN-based methods and are hence, computationally more expensive [19]. This makes them harder to train, and unsuitable for easy deployment and real-time inference.

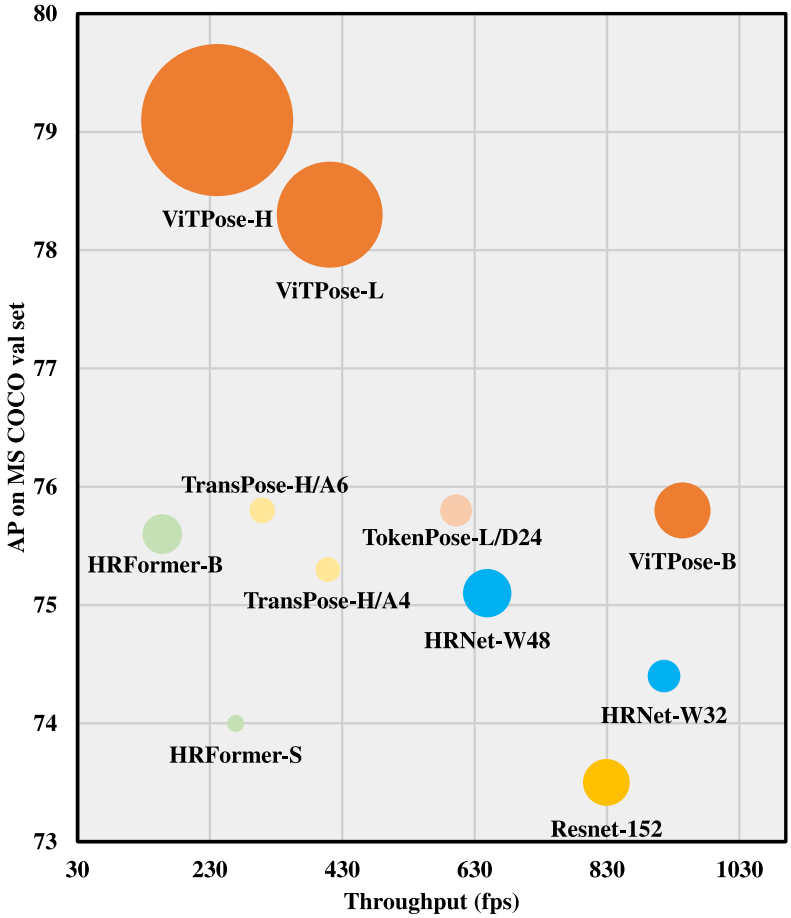


Figure 2.6: Comparison of ViTPose (in orange) and other SOTA methods on MS COCO val set regarding model size, throughput, and precision. Throughput is calculated by dividing the number of images processed by the processing time. The size of each bubble represents the number of model parameters [92].

OpenPose, introduced in [6] is one of the most popular bottom-up pose estimation methods that

use a non-parametric representation, called Part Affinity Fields (PAF). PAF is a set of 2D vectors that indicate the degree of association between the body parts [59]. For an input image, this method first calculates a set of PAF and part confidence maps that represent the location of each joint (see Fig. 2.7). Later, a matching algorithm called bipartile matching is utilized for forming associations between the predicted joints. Full-body poses are then assembled for all people in the image. Cheng *et al.* [10] later introduced HigherHRNet to address the problem of scale variance in bottom-up HPE. It generates a high-resolution feature pyramid with multi-resolution supervision and heatmap aggregation in the training and inference stages respectively to generate scale-aware high-resolution heatmaps. It outperforms all other bottom-up methods especially due to the ability to detect small and medium-sized persons in the image [10].

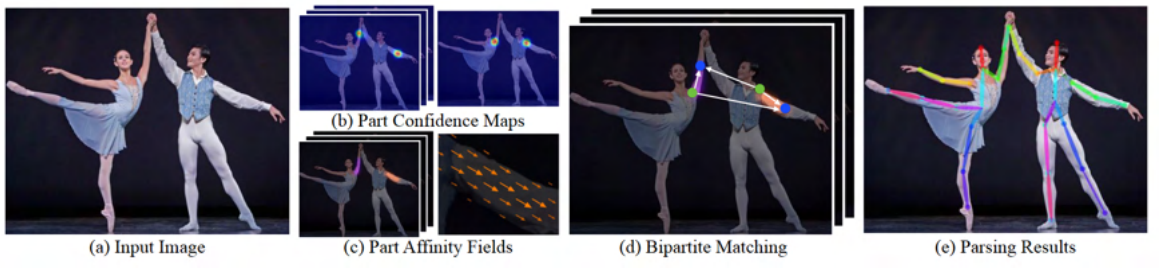


Figure 2.7: OpenPose workflow [6]

YOLO-Pose [55] is a heatmap-free approach for 2D HPE based on the popular YOLOv5 object detection framework. It was the first research to unify the field of multi-person HPE and object detection since major challenges like occlusion, and scale variation are common to both. Thus, any advancements in the field of object detection will directly benefit the approach of YOLO-Pose. It jointly detects bounding boxes for multiple persons and their corresponding 2D poses in a single forward pass. It is, to some extent also capable of predicting a keypoint outside the predicted bounding box, which is a common problem with top-down approaches. It uses the CSPDarknet53 [90] as the backbone and PANet [47] for fusing features of various scales from the backbone. YOLO-Pose also extended the idea of the IoU loss from bounding box detection to keypoint detection based on the OKS metric (see Sec. 2.3.2). OKS is not only used for evaluation but also as a loss during training. It is a scale-invariant loss that weighs each keypoint differently.

The HPE model by YOLOv7 is based on YOLO-pose and achieves SOTA real-time pose estimation result [89]. YOLOv7-pose replaces the backbone architecture of YOLO-pose with its own and utilizes many anchor boxes for multi-scale detection. The anchors in YOLOv7-pose are associated with the feature maps at 1/8, 1/16, 1/32, and 1/64 of the image size respectively. It then uses a similar method as YOLO-Pose for the fusion of these features at different scales and also for predicting boxes and keypoints. For each anchor, the keypoint head predicts 51 elements (x coordinate, y coordinate, and confidence for each of the 17 keypoints) and the box head predicts six elements (class, x coordinate, y coordinate, width, height, and confidence). The architecture of YOLOv7-pose is shown below (Fig. 2.8).

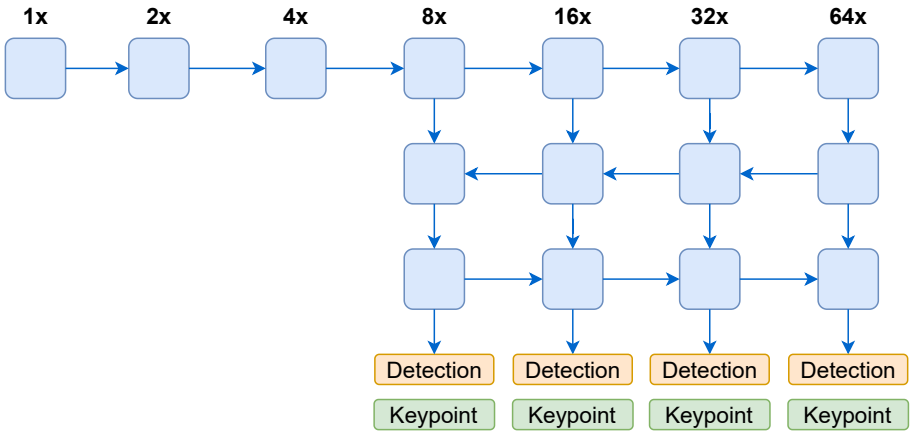


Figure 2.8: YOLOv7-pose architecture [89]

YOLOv8[5] introduced six different HPE models based on their object detection architecture. It uses a spatial pyramid pooling [32] block to combine features from high-resolution and low-resolution layers enabling accurate pose estimation for both small and large objects. YOLOv8 claims to be even faster, simpler, and more accurate than the previous YOLO versions. YOLOv8 has also focused on reducing the number of parameters in their model and achieving faster inference (see Fig. 2.9). However, since both YOLOv7 and YOLOv8 have been proposed very recently, there are no detailed comparison studies yet. The comparison of the accuracy of most of these models on the COCO Keypoint dataset[6] is shown in Table 2.1. The exact accuracy of YOLOv7-pose on the dataset is not provided in the paper.

[5]YOLOv8 GitHub: `https://github.com/ultralytics/ultralytics`
[6]COCO Keypoints val2017: `https://cocodataset.org/#download`

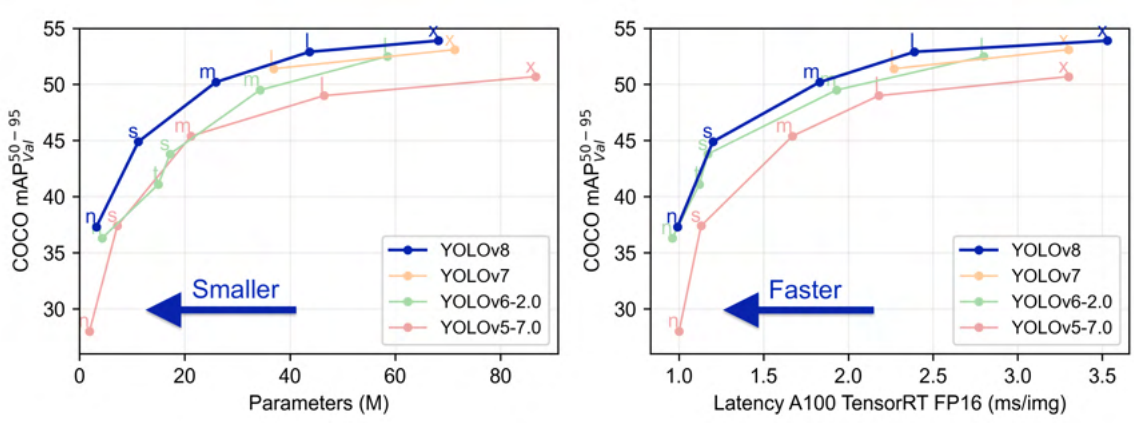


Figure 2.9: Accuracy of YOLO versions on COCO dataset plotted against Number of Parameters (left) and Latency (right). Latency measures the inference time (ms) required to process a single image [7]

Table 2.1: Performance comparison of the recent HPE methods on the COCO keypoint dataset

HPE Method	Pose mAP_{50}	Pose mAP_{50-95}
OpenPose [6]	84.9	61.0
HigherHRNet [10]	88.2	68.4
YOLO-pose (YOLOv5l6-pose) [55]	90.2	69.4
YOLOv8x-pose-p6[8]	91.2	71.6
ViTPose-B[9] [92, 54]	95.0	81.1

2.3.4 In-Bed Pose Estimation

Although there are several methods developed for HPE, very few address the resting horizontal position of a human. For solving this problem of in-bed or sleep pose estimation, many methods rely on using multi-modal data. Pressure mapping has been extensively used for 2D and 3D in-bed HPE [17, 13, 7]. These methods need sophisticated equipment which restricts their deployability and they still suffer from the problem of free-moving limbs producing negligible contact pressure [13, 39]. Clever *et al.* [12] used depth images to successfully estimate the 3D pose and pressure image under the cover. They use their synthetic dataset, BodyPressureSD [11] to train their model. Achilles *et al.* [2] inferred 3D patient pose from depth video in real-time.

[7] Source: `https://docs.ultralytics.com/models/yolov8/`
[8] YOLOv8 GitHub: `https://github.com/ultralytics/ultralytics`
[9] Using detection results from a detector that obtains 56 mAP on person class.

Kyrollos *et al.* [38] used both pressure mapping and depth data to train a vision transformer for under-the-cover infant pose estimation. Some other methods [50, 51, 58] used different types of Infrared (IR) imaging techniques to obtain more information about the person under the cover. Only Liu *et al.* [49] used RGB images for in-bed HPE, but only the cases with little-to-no occlusion were considered.

2.3.5 Datasets in 2D Human Pose Estimation

There are many different datasets available that are commonly used for training and evaluating HPE models. The earlier datasets like LSP [36] and Penn Action [97] are small-scaled and only deal with single-person scenes. More recent datasets like COCO [46], MPII [4], CrowdPose [42], and PoseTrack [3] are large-scaled datasets widely used for multi-person HPE. COCO dataset is the largest multi-person HPE dataset which is frequently used for benchmarking. MPII is a 2D HPE dataset that contains “in-the-wild” and indoor green screen annotated images of people doing around 500 different activities. Human3.6M [35] is the largest single-person 2D/3D HPE dataset containing video sequences of 11 actors performing 15 different possible activities.

There exist some datasets that specifically deal in HPE with occlusion like OCHuman [96] and Crowdpose [42]. OCHuman contains 4731 images of 8110 humans with bounding-box, human pose, and instance mask annotations. Crowdpose contains 20,000 images in total and 80,000 human instances with the aim to improve performance in crowded scenarios as the name suggests. The images in these datasets only represent self-occlusion and persons occluded by other persons. It does not take into account extreme occlusion by other objects in the scene. Furthermore, they still consider only ’in-the-wild’ situations where the pose distribution and camera viewpoints are quite standard.

For the case of in-bed or sleep pose estimation, there are not many publicly available datasets. Liu *et al.* [51] introduce an in-bed pose estimation dataset created with one male and one female realistic life-size mannequins in a simulated hospital room. They use an Infrared Selective (IRS) image acquisition system but also provide high-res RGB images. Their dataset does not include any external occlusions on the mannequins except the hospital gowns. The only dataset that can potentially serve as a benchmark for in-bed HPE is the Simultaneously-Collected Multimodal Lying Pose (SLP) dataset [48]. SLP provides RGB, Long Wavelength IR, pressure mapping, and depth imaging modalities taken from an overhead angle. They also provide the same poses in three different cover conditions: no cover, a thin sheet with $\approx$ 1 mm thickness, and a thick blanket with $\approx$ 3 mm thickness (see Fig. 2.10). The main dataset is recorded in a home setting with 102 participants and a test set in a hospital room setting with 7 different participants. There are a total of 14715 samples in the dataset, making it the only large-scale dataset for in-bed HPE. The participants were asked to lie in natural poses in supine, left, and right side postures. For each posture and cover condition, 15 images were taken in 4 modalities.

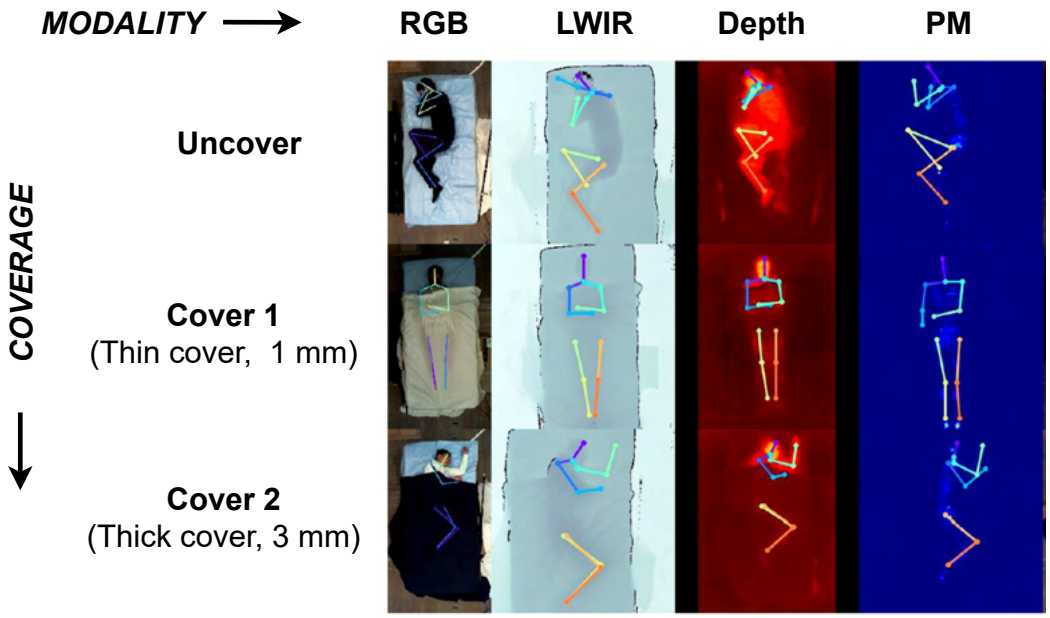


Figure 2.10: SLP image samples showing different modalities and cover conditions [48]

2.4 Synthetic Data

The emergence of CNNs and other deep learning-based methods in CV has created a need for large amounts of visual data for training and testing. The traditional methods to obtain this data require obtaining real data, followed by a difficult process of manual annotation. This process proves to be costly for large-scale datasets. The difficulty of annotating also increases drastically with the need for complex, detailed, and accurate annotations. The accuracy of these annotations cannot be guaranteed due to human error. In some situations such as crowd counting or occluded keypoints, accurate annotation may not even be possible for a human. For the task of this thesis, obtaining accurate annotations of the human pose under heavy occlusion is a challenging task, not only for a machine but also for a human [48] (See Appendix, Fig. 4). Other concerns regarding privacy and legal restrictions also hinder the generation of real datasets. On the contrary, the synthetic data generation process can be automated with minimal effort and large-scale datasets can be generated in a relatively short time without privacy and legal concerns. Furthermore, the annotation process can also be automated to produce detailed and accurate ground truths. Synthetic data can also be useful in domains where collecting real data is not possible at all due to physical or safety restrictions. It can also simulate rare or dangerous situations. In certain cases, it is often necessary to build a completely new dataset due to the specificity of the application. Synthetic datasets have proven to be effective in tackling such problems [62].

There also exist certain limitations of synthetic data. Real data is often taken as a reference to generate synthetic data and the inherent biases present in the reference data are replicated in the synthetic data [56]. Furthermore, the models trained on synthetic data often fail to perform

well in real-world tasks due to domain gap. Domain gap refers to the discrepancy between the simulated or synthesized data and real-world data. This gap arises because of the inability of synthetic data to accurately represent the complexity present in the real world.

2.4.1 Popular Synthetic Datasets in Computer Vision

There are many types of synthetic datasets in CV. Synthetic composite datasets are made by augmenting or digitally manipulating real image data. SURREAL [85] is the first such large-scale synthetic dataset containing virtual video animations. It is created using lab-recorded MoCap data that provides depth, body parts, optical flow, 2D/3D pose, and surface normals ground truth. Synthesized 3D humans were projected onto real background images in this case (see Fig. 2.11). CNNs trained on SURREAL dataset achieved accurate human depth estimation and human part segmentation in real RGB images [85]. Foggy Cityscapes dataset [76] for semantic segmentation and object detection was created by generating synthetic fog on real images from Cityscapes [14]. In such cases, the real environments and objects are still so important that the creation of completely synthetic counterparts is not worth the efforts [56]. Training with the Foggy Cityscapes dataset significantly improved the performance of SOTA CNNs for semantic foggy scene understanding on real data [76]. KITTI-360 dataset [1] augmented the real-world KITTI dataset [27] by placing high-quality vehicle models into existing KITTI scenes with realistic lighting conditions.

Figure 2.11: Examples from SURREAL dataset [85]

Virtual synthetic data refers to a dataset that is completely synthesized without any real elements. Virtual-KITTI [24] is an example of such a dataset used in the field of automated driving. It is a synthetic recreation of a part of the KITTI dataset. The authors proved that pre-training on synthetic data can boost performance after fine-tuning. Another synthetic dataset for HPE, SynPose300 [77] is frequently used to test the robustness of 3D HPE against viewing distance and angle, larger clothes, and difficult actions. SynPose [94] is a large-scale HPE dataset specifically designed for crowded HPE in classroom and meeting scenarios. The SYNTHIA dataset [72] contains photo-realistic frames rendered from a virtual world with precise pixel-level annotations for semantic segmentation in urban scenarios. The inclusion of the SYNTHIA dataset in the

training of CNNs significantly improved the performance on the semantic segmentation task [72].

2.4.2 Reducing the Domain Gap

There are various methods implemented for reducing the domain gap between synthetic and real datasets. Some methods have tried to reduce this gap by improving the realism of the synthesized data by rendering 3D face models with an unprecedented level of realism and diversity [91] or by using unlabeled real data using a GAN-like network [79]. Others [25] have tried to force the networks to ignore differences between the synthetic and real domains. Domain Randomization (DR) is one of the most promising approaches to enable direct transfer learning from a synthetic-to-real domain, also known as *Sim2Real transfer* [62]. Ideally, the distribution of synthetic data should be exactly equal to the real data or at least cover it entirely. But this is rarely feasible. DR is an approach that intentionally abandons photo-realism in the images by randomly perturbing the virtual scene [82]. Some non-photo-realistic elements are added either in the objects or environment that forces the network to learn only the essential features of the image [84]. Furthermore, forcing the synthetic data to represent these features in a photo-realistic way will only prove to be more expensive for the data generation process. Tremblay *et al.* [84] used DR to generate synthetic data for object detection resembling the KITTI dataset [27]. 3D models of the objects of interest were placed in a 3D scene with many other objects that were not of interest. Random colors and textures were applied to all models along with random lighting in the scene. A CNN-based model with COCO [46] weights was trained on this DR-based dataset as well as the Virtual-KITTI dataset [24]. Training with their DR-based dataset resulted in a better performance than training with the Virtual-KITTI dataset. They also prove that a better result can be achieved by fine-tuning this network with real data than using real data alone.

Domain Adaptation (DA) is a set of techniques used to make a model trained on a source domain perform well on a different target domain. In *Sim2Real* transfer, the source domain is synthetic and the target domain is real. Approaches for DA can be broadly classified into two categories: data-level and model-level. The data-level approaches directly manipulate the synthetic data to make it work better on real data whereas the model-level approaches change the model, its feature space, or the training process to obtain the same goal. The data remains untouched in the model-level approaches. Fig. 2.12 illustrates the domain gap problem in classification tasks and the use of DA techniques for aligning feature distributions of source and target domains resulting in a correct classification. Generative Adversarial Networks (GAN) [31] and methods based on them are also widely used for DA or smart manipulation of real data to include some nontrivial transformations. Srivastava *et al.* [79] presented a GAN-based approach for gaze estimation that refines the synthetic eye images to make them more realistic. In the field of face recognition, Huang *et al.* [34] presented TP-GAN that can generate a frontal view of the face of a person given an image of the face.

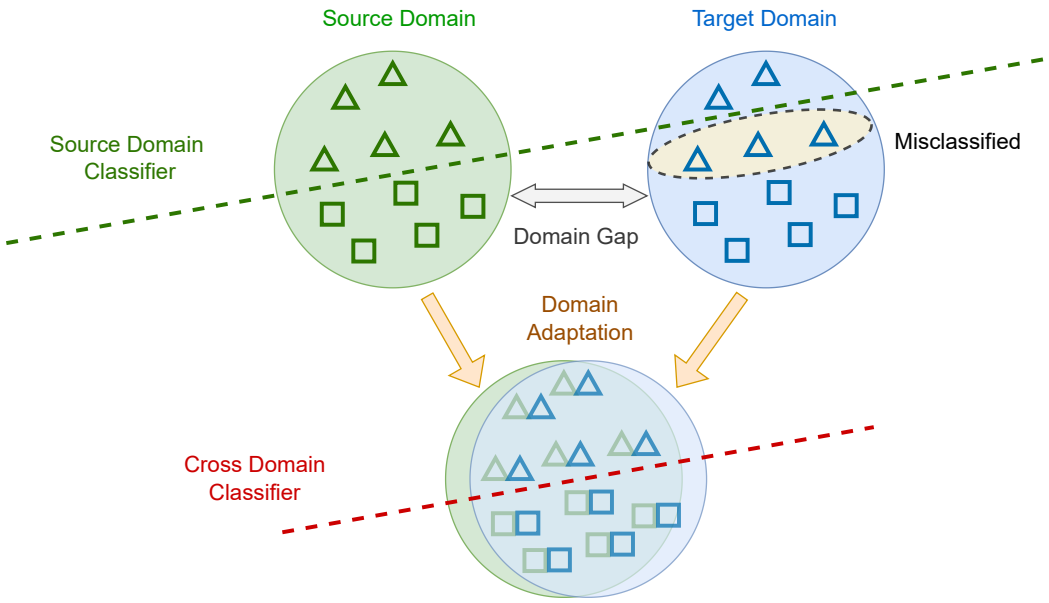


Figure 2.12: Working of domain adaptation techniques. Source and target feature distributions (top), and new feature distributions after domain adaptation (bottom) are shown. Reduction in the domain gap results in correct classification [44].

3 Adapting the Synthetic Dataset

This chapter focuses on refining and adapting the existing synthetic dataset to achieve the goal of improving human pose estimation in challenging scenarios. Firstly, the initial state of the synthetic dataset is described. Next, the real dataset to be used as the counterpart to the synthetic dataset is introduced. This is important for evaluating the effectiveness of the synthetic data. In the subsequent sections, the specific modifications made to the synthetic datasets will be explored. The decisions and methods used to enhance the quality of the dataset will be explained in detail.

3.1 Previously Developed Synthetic Dataset

The initial version of the synthetic dataset was generated in Blender[1] during the internship period. An approach similar to the SURREAL [85] dataset was chosen. It was a composite dataset with 3D synthetic humans projected on a real background (see Fig. 3.2). The backgrounds of the scene consisted of real photo-based HDRI images. Several rest poses were manually created to simulate a person lying on a bed in supine, left, right, and downward positions. A soft-body simulation of cloth falling on the horizontally lying person was performed to introduce a blanket-based occlusion in the dataset. The blanket was made to fall over the person at a certain angle to simulate some creases and wrinkles (See Fig. 3.1). Blanket-falling simulation was used instead of low-level geometric shapes or other objects as occlusions as the soft-body simulation would generate more contours roughly representing the pose of the person underneath. This was intended to help the model effectively learn and in the inference stage, make an educated guess about the pose distribution under such occlusion.

Images were then rendered from a virtual camera inside the scene whose position was partly randomized. The ground truth consisted of a bounding box as well as 17 keypoint annotations according to the COCO keypoint annotation format [2]. The 17 keypoints are namely Nose, Eye-L, Eye-R, Ear-L, Ear-R, Shoulder-L, Shoulder-R, Elbow-L, Elbow-R, Wrist-L, Wrist-R, Hip-L, Hip-R, Knee-L, Knee-R, Ankle-L, Ankle-R. Here, L and R stand for the left and right sides, respectively. These annotations were exported in COCO-like *JSON* file format and also in YOLO format. The YOLO format consists of a *txt* file containing annotations for each person on a separate line. There are only two minor differences between the actual values of the ground truth. First, the COCO format uses absolute pixel locations of the keypoints and bounding boxes whereas the YOLO format uses normalized values. And second, the bounding box annotations of the COCO

[1] Blender 3.5.1: an open source 3D creation suite available online at `https://download.blender.org/release/`
[2] See `https://cocodataset.org/#keypoints-eval`

format are *xywh* style where *x* and *y* are the coordinates of the top left corner and *w* and *h* are the width and height of the box. For the YOLO format, the *x* and *y* denote the coordinates of the center of the box.

Figure 3.1: Close-up example of an image in the dataset showing the blanket simulation

Figure 3.2: Previously developed synthetic dataset

3.2 Real Dataset

To effectively analyze the quality of a synthetic dataset, it is essential that a model trained on the data also performs well in real-world scenarios. Hence, a real dataset as a counterpart to the synthetic dataset is needed. The SLP dataset [48] presented in Sec. 2.3.5 offers a very unique solution to the problem of sleep pose estimation. The dataset consists of a large number of

people which increases its diversity substantially. The presence of three different cover conditions enables testing the models independently for each setup. It also facilitates a study about the accuracy of a model on the RGB images and how it is affected by the thickness of the cover. It also presents a large amount of sleeping pose diversity. Many extreme poses are represented in the dataset which makes it quite suitable for use in training and testing HPE models (see Appendix, Fig. 1). However, it lacks diversity in terms of colors, textures, backgrounds, and lighting conditions. All the images in the SLP dataset represent a controlled lab environment. The elements like bed, blanket, bedsheet, and floor are the same in almost every image of the lab environment. The same is also valid for the hospital room environment. Furthermore, the camera viewpoint is strictly overhead with the person or the bed positioned in the same place with respect to the camera. The lighting conditions are also very uniform with the absence of any variations in the intensity, color, or position of lights. These factors do not make it an ideal dataset for training a robust model that can generalize to different real-world images. Evaluating any models trained on the SLP dataset and also validating and testing them on images from the same SLP dataset can lead to a bias in results. This is usually also observed in datasets with high diversity but to a much lesser extent. Unfortunately, due to the lack of any other potential datasets for this thesis, the SLP dataset was chosen as the real dataset for the testing and training process.

The SLP dataset contained two different environments, namely home, and hospital with 102 participants in the former and 7 in the latter. As the environment setting is inconsequential for this thesis and all the other factors were identical in these two setups, both subsets were combined. Since the focus of this thesis is to try to tackle the challenge of sleeping poses and occlusion only with RGB images, only this modality from the SLP dataset is considered (see Fig. 3.3). The SLP dataset focuses on major limb annotations and follows the joint definition of [36] with 14 joints. However, many major HPE methods use the COCO format for keypoint annotations which contain 17 keypoints as mentioned in Sec. 3.1. The difference lies in the facial keypoints where the SLP dataset provides only 2 keypoints (Thorax, and Head Top) instead of 5 in COCO.

Figure 3.3: SLP image data samples from in-bed supine (left) and side postures (right) in RGB modality with three cover conditions [48]

This discrepancy in the annotations needed to be fixed to use both datasets in the study. Since most of the methods of HPE use the 17-keypoint annotation format, the SLP dataset has been re-annotated. Although the SLP dataset contains images of persons in three cover conditions, in

all of these images, the person was only covered at the most till the neck. Hence, the face was always left uncovered. The latest HPE model, *yolov8x-pose-p6*[3] was used to run inference on all the images in the SLP dataset to check its suitability for re-annotation of the facial keypoints. Only the five relevant facial keypoints from the inference of the model were extracted and the remaining keypoints from the original annotations were retained to form the re-annotated SLP dataset. The bounding box annotations were also retained from the original SLP dataset. The re-annotated SLP images were visually checked for any errors in the facial keypoints. The number of wrongly annotated images was small (< 1%) due to the occlusion-free faces of the persons. These images were excluded from the dataset. This was of no particular concern due to the large number of images in the original SLP dataset. Some examples from the re-annotated SLP focusing on the facial annotations are shown in Fig. 3.4.

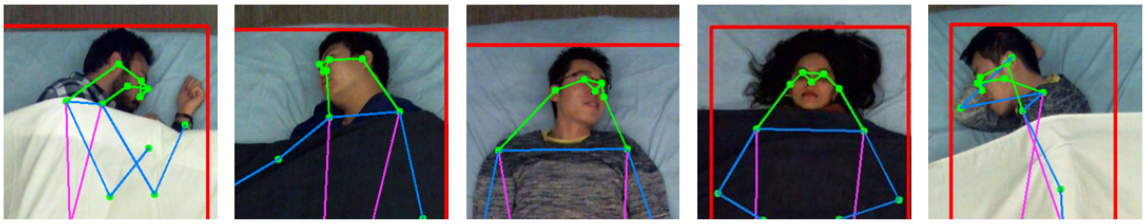


Figure 3.4: Cropped image examples from the re-annotated SLP dataset showing new facial annotations

3.3 Modifications to the Synthetic Dataset

The first version of the synthetic dataset was only intended to verify the capability and accuracy of the data generation pipeline concerning character manipulation, blanket simulation, realism of the data, and automatic annotations. It had none of these qualities of an effective synthetic dataset. Nowruzi *et al.* [63] provide insights about training deep networks using synthetic datasets while also comparing the effects of having a limited amount of real data. They conclude that the diversity of the synthetic data is more important than photo-realism. Similarly, Fabbri *et al.* [21] also prove that diversity in terms of people's appearance is crucial for bridging the domain gap. Hence, a series of modifications were needed to make it suitable and effective for training deep learning-based methods. These modifications were undertaken in the scope of this thesis and are described in this section.

3.3.1 Bridging the Reality Gap with Domain Randomization (DR)

The main aim of the modification process is to reduce the domain gap between the synthetic and the real data. This work is built on the DR approach described in Sec. 2.4.2. The non-essential elements in the scene of synthetic data were identified namely, background, lights, blanket,

[3]YOLOv8 GitHub: `https://github.com/ultralytics/ultralytics`

character's skin, hair and clothing, ground plane, and bed. Various properties of these objects were then randomized. This approach is intended to add the diversity in terms of colors, textures, lighting, etc. that is missing in the synthetic dataset. Many different real-world photo-based HDRI backgrounds are available at Poly Haven[4]. These backgrounds, when imported in Blender, have integrated lighting according to the scene. The complex integrated lighting recreates many forms of lighting present in natural or artificial settings in the real world. Hence, a wide variety of backgrounds representing different conditions such as indoor, outdoor, and studio settings, high contrast and sharp lighting, etc. were chosen. Some of these are shown in Fig. 3.5. To introduce more randomness, the HSV values of these images were randomly chosen. Also, the orientation of the image and the intensity of the background lighting were randomly set. The lower intensity of the lighting simulated dimly lit scenes.

Figure 3.5: Examples of the HDRI backgrounds showing diversity in color, lighting, location, and complexity

Figure 3.6: Examples of the materials used for the clothes showing diversity in color, pattern, texture, and reflectivity

The next elements to be targeted for modification were the clothes in the scene. This included the clothing for the character and the blanket in the scene. Each character by default had realistic clothes and materials. Since most of the characters would be covered with the blanket, the materials of the clothing were not of great significance. Hence, only the colors of the clothing

[4]Poly Haven HDRIs: `https://polyhaven.com/hdris` with CC0 license: `https://creativecommons.org/publicdomain/zero/1.0/`

were chosen to be modified. The HSV values were randomly chosen between 0 and 1. The same was also done for the character's hair and skin. Since the blanket was the core element of the synthetic dataset, its materials were of much significance. A variety of materials representing a diverse range of colors, roughness, reflectivity, patterns, and textures were obtained from Blenderkit[5]. Some of these materials are shown in Fig. 3.6. The resulting dataset after DR can be seen in Fig. 3.7.

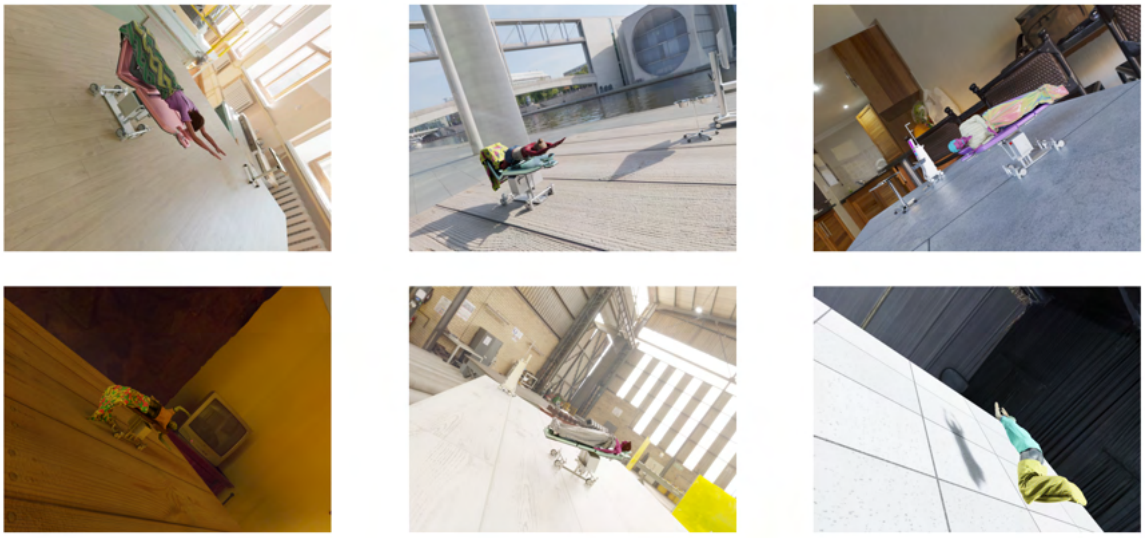


Figure 3.7: Examples from the synthetic dataset after domain randomization

3.3.2 Virtual Camera Positions

Initially, the positions of the virtual camera were randomly sampled from a hemispherical surface in the scene. The orientation of the camera was set such that it roughly pointed toward the character in the virtual scene. This was achieved by making the camera point towards a point sampled randomly from an imaginary cuboid around the character in the scene. However, the characters in the images were too small and far from the virtual camera. The camera placement was then changed to generate images from an overhead angle to simulate the camera placement in the SLP dataset. This was done to limit the scope of the study to the use and effectiveness of synthetic data for occluded in-bed HPE. Other factors such as a non-standard camera perspective were ignored due to the lack of comparable real datasets representing such factors. Since the characters appeared only from an overhead viewing angle, the ground plane of the scene was removed. Hence, the HDRI background was now directly visible in the scene. The ground plane, as opposed to the HDRI backgrounds, lacked complexity.

[5]Blenderkit: a free library of Blender models, materials, HDRs, scenes, and brushes. Site: `https://www.blenderkit.com/`

3.3.3 Increasing the Pose Diversity

The number of extreme and uncommon poses is high in the SLP dataset. Hence, the network needs to learn a wide representation of the complex poses of humans under the cover to perform effectively on the SLP dataset. Purkrábek *et al.* [65] address the challenges in HPE posed by extreme viewpoints and poses by introducing a new method for synthetic data generation - *RePoGen*, RarE POses GENerator. They generate extreme poses, even if they occasionally deviate from anatomical accuracy while also allowing some intersection of body parts with each other. They demonstrate that infeasible poses or small-scale mesh intersections do not impede the training. For generation of poses, each body angle is sampled from a distribution. They compare two distributions, namely uniform distribution and baseline distribution. Uniform distribution produces more frequent extreme poses. The baseline distribution is an asymmetric normal distribution, composed of two normal distributions with different variances (see Fig. 3.8). The comparison between these distributions conducted by the authors was inconclusive with the uniform distribution performing better on bottom-view images and the baseline distribution performing better on top-view images of upright humans.

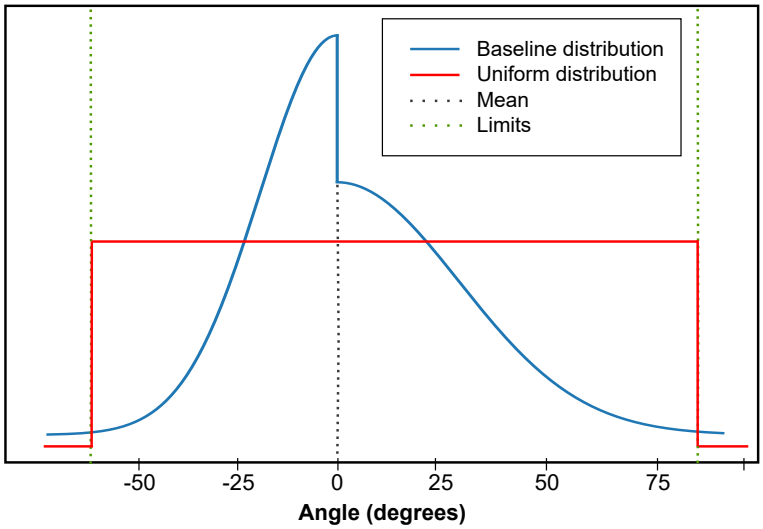


Figure 3.8: Examples of joint angle distributions used in data generation of *RePoGen* [65]

In the first version of the dataset, the poses of the characters were manually created and were limited. They did not represent the complexity and diversity of poses that a real human can achieve. Hence, to introduce some randomness and diversity of poses in the synthetic data, the joint rotations of the character models along all three axes needed to be varied. For this, several joints were chosen, namely Neck, Shoulder-L, Shoulder-R, Elbow-L, Elbow-R, Wrist-L, Wrist-R, Hip-L, Hip-R, Knee-L, Knee-R, Foot-L, Foot-R, Lower Spine, Upper Spine, and Pelvis. For introducing more extremities and variation in the poses, two methods were considered taking some elements from *RePoGen*:

- The first method utilizes the realistic poses created in previous work. After applying this

pose to a character, each of the joints mentioned above is rotated by an angle along each axis in each frame of the scene. The angle is uniformly sampled from a range of -5 to 5 degrees (see Fig. 3.9). The blanket simulation is performed simultaneously. The dataset generated with this method is henceforth called **RealPose**. The example images are shown in Fig. 3.12.

- In the second method, the base pose is also randomly generated. For all the joints mentioned above, specific lower and upper limits for the joint angles were chosen for each axes. These limits were decided by varying each joint angle independently until it reached an approximate limit of a human body. Unlike in *RePoGen*, the limits for each joint are not centered at 0 for the characters due to their fundamental definition of joint axes. Knowing the exact angle limits of human motion was unnecessary because unrealistic poses do not affect the model's training [65]. Over-engineering of the base poses may result in more bias in the data generation parameters. Each of the joints was then rotated every frame in the same manner as the previous method. The dataset generated with this method is henceforth called **RandPose**. The example images are shown in Fig. 3.13.

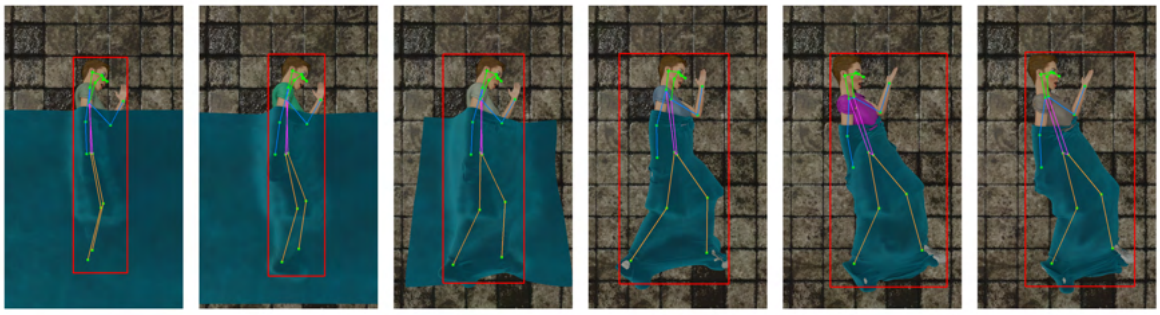


Figure 3.9: Character with joint angle variation of maximum 5 degrees in each frame (shown with annotations and blanket simulation)

The baseline or asymmetric normal distribution introduced in *RePoGen* also presented an interesting option to generate the base poses. A conventional normal distribution was first chosen to generate some poses for evaluating the suitability of the distribution. The mean was set to the midpoint of the joint angle limits. Many different variances were tried, but the ideal variance was not found during the implementation. Setting a small variance resulted in very few extreme poses of the character. This made it similar to the RealPose dataset in terms of the extremity of poses. On the other hand, setting a large variance resulted in many extreme poses, but with a preference for the "mean" pose. The mean pose is the pose formed by the mean values of each of the joint angles (see Fig. 3.10). Many of the poses represented some variations of the mean pose of the character, hence lacking the needed diversity. Also, the asymptotic form of normal distribution resulted in too many poses with quite extreme joint angles that fall considerably outside the human limit. It created infeasible poses with a high level of mesh intersection. These ill-suited characteristics and the inability to find the right variance for each joint discouraged the use of normal and normal-like distributions for this thesis.

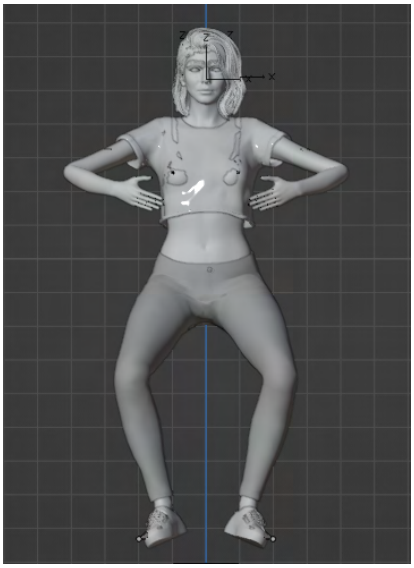


Figure 3.10: Mean pose of the character formed by the mean joint angles of the normal distribution

Figure 3.11: Examples of sampled base poses in the RandPose dataset (shown without the blanket for ease of visibility)

The RealPose dataset has less pose diversity than RandPose. However, the pose generation method of RandPose made the blanket simulation more challenging. As the base pose of the character can frequently result in an extreme pose, problems in the soft-body collision happening along with the joint movement in each frame resulted in the blanket slipping or bouncing away from the leg of the character in many instances. In a few images, the legs of the characters penetrated through the blanket because of the extreme lifting of the legs from the hips in the base poses. As the sole aim of the second method is to introduce more extreme poses, this issue could not be fixed by simply eliminating these poses. These problems with the blanket were present in roughly at least 15%-20% of the generated images. Some example images showing the problem with the blanket simulation are presented in the Appendix, Fig. 6. The partial occlusion and abnormalities caused by such unintended blanket simulation could be considered as yet another

unrealistic random element in DR. Hence, the dataset was kept as it is and further comparisons between the datasets are shown in the Sec. 5.4 and 5.6.

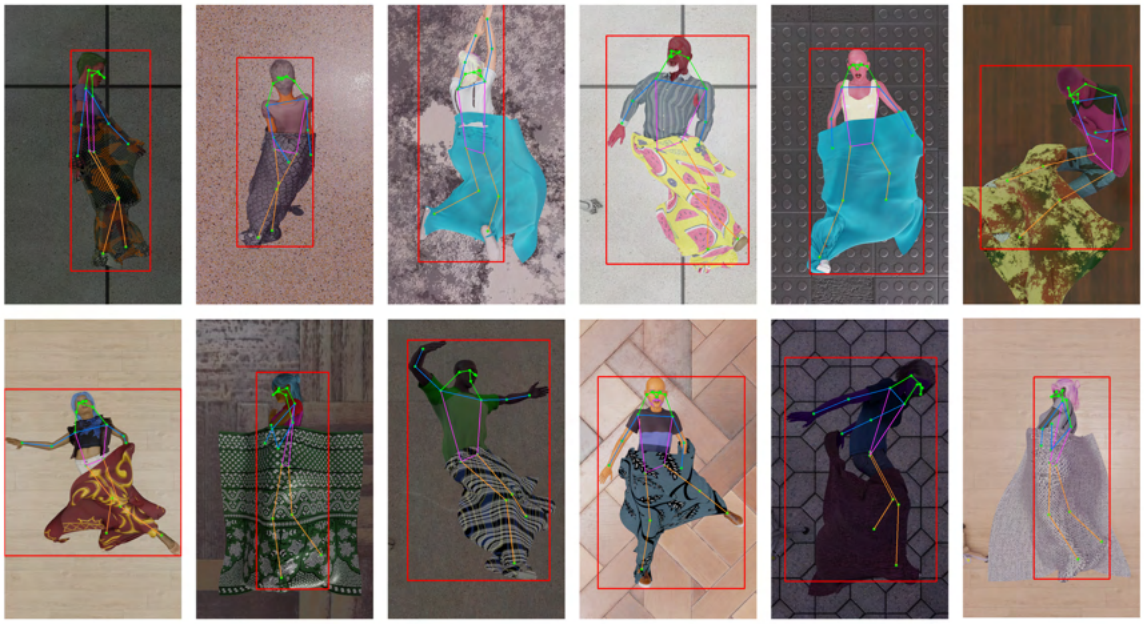


Figure 3.12: Examples from RealPose dataset

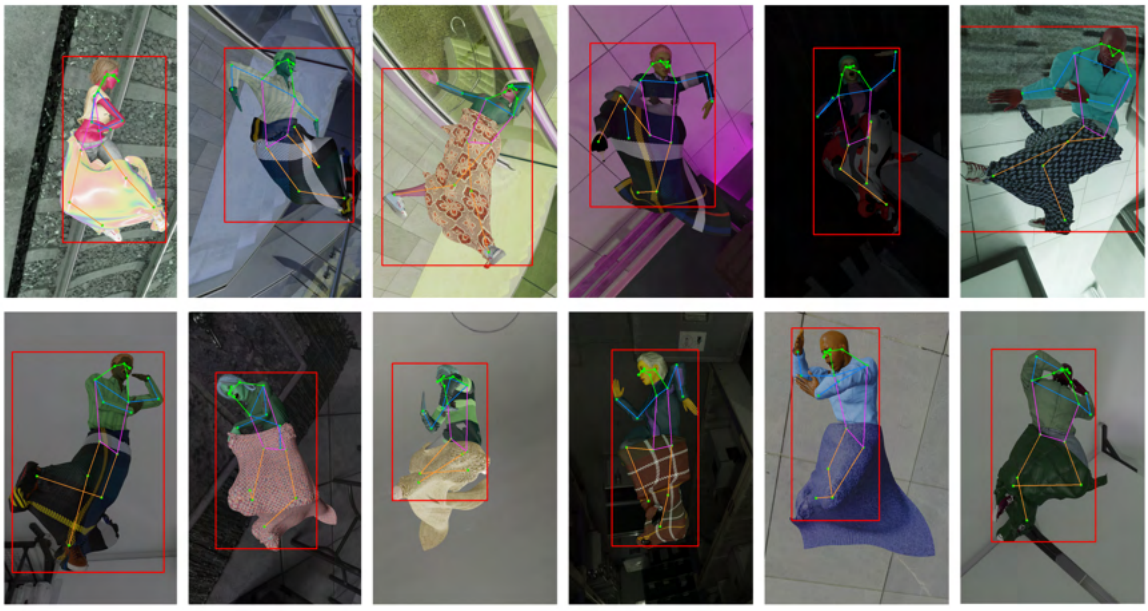


Figure 3.13: Examples from RandPose dataset

4 Methodology

In this chapter, the experimental setup and the important aspects of the training process with the synthetic dataset are described. The training methodology and the evaluation metrics chosen for assessing the efficacy of the models are detailed. It also explains the reasoning behind the choice of models used for training with synthetic data. The preparation of the dataset and the changes made to the models are then described. Additionally, the selected DA technique's working and integration within the models are presented in detail.

4.1 Model Selection: YOLOv7-pose and YOLOv8-pose

Choosing an effective model for pose estimation of humans in horizontal positions, particularly in the presence of occlusions such as blankets, necessitates certain key features. A successful model should show robustness to occlusion, capturing nuanced details even when body parts are imperceptible because of the blanket. The ability to generalize well to diverse human sleeping positions and various real-world complexities like lighting conditions and complicated environments is also important. In addition, the model should be adaptable and easily trainable for various tasks. Additionally, it should be computationally inexpensive with fast inference for easy deployment.

Bottom-up approaches of HPE, especially in the scope of this thesis, do not offer substantial advantages to ignore their shortcomings in accuracy. In this scenario, due to severe occlusion of the body joints, accurate prediction of the keypoint location at a local level becomes a tough task. Without a global context, it's hard to identify joints using bottom-up methods, which are disadvantaged by this limitation. Top-down methods, on the other hand, start the process of HPE by detecting the person in the image and then estimating the location of individual body parts, allowing for a holistic understanding of the overall poses. The coarse-to-fine approach of the top-down methods is more suitable for piecing together the obscured body parts by focusing on local details and refining its predictions based on the global understanding of the scene.

Furthermore, the main disadvantages of the top-down methods as discussed in Sec. 2.3.1 are the speed of inference and reliance on a robust prediction from the human detector. The inference speed is mainly affected in crowded multi-person HPE. Since this thesis focuses on improving the models' performance on the SLP dataset containing only a single person in each image, this disadvantage is inconsequential. The bottom-up methods like OpenPose [6] and HigherHRNet [10] are more suited for crowded pose estimation with high human-based occlusion. The current SOTA transformer-based architectures like ViTPose [92] have great potential but are hard to train

and require a large amount of data. Also, it relies on a good object detector as a prerequisite for HPE which necessitates the use of models like YOLO, SSD, etc.

For these reasons, the latest HPE methods based on the YOLO series, namely YOLOv7-pose [89] and YOLOv8-pose are chosen. As opposed to ViTPose, these models provide an end-to-end solution for HPE. Moreover, both of these methods claim to be faster than real-time with a lower model complexity while still maintaining good accuracy. Another advantage of these models is the continuous research and improvement into the YOLO series with a focus on speed and accuracy. It has now also expanded to other challenges in CV such as image segmentation and pose estimation. The strong and efficient backbones of the YOLO models are also advantageous for learning the pose representation under the blanket. YOLOv7-pose offers one pre-trained model, *yolov7-w6-pose* whereas the YOLOv8-pose method offers six. The largest and the most accurate model, *yolov8x-pose-p6* is chosen for further experimentation. These models are hereon just referred to as YOLOv7 and YOLOv8 respectively. All the pre-trained models provided are trained on the COCO keypoint dataset.

4.1.1 Modifications in the Models

Both the HPE methods chosen have provided the official implementation based on PyTorch on GitHub[1,2]. Some minor changes are made to the code for the YOLOv7-pose and YOLOv8-pose models. To take the TL approach, the ability to freeze certain layers was integrated into the models. The parameter *freeze* was added to the input arguments for YOLOv7-pose indicating the number of layers whose weights are to be frozen. In YOLOv8-pose, a callback function was added to achieve the same result. YOLOv8-pose reports and logs the precision, recall, mAP_{50} and mAP_{50-95} values for both the bounding box as well as pose estimation during the testing phase. This is not the case for YOLOv7-pose and hence, it was added in the code with the same computations that are done in YOLOv8-pose. This was done to facilitate a comparison between the models with the same metrics. Furthermore, for testing these models after the training, only the weights from the best-performing epoch were chosen. By default, this best-performing epoch was determined by a combination of the mAP_{50} and mAP_{50-95} values for the bounding box. This was changed to a combination of the newly added mAP_{50} and mAP_{50-95} values for pose estimation to get a more relevant evaluation of the model. The weight for pose mAP_{50} was set to 0.75 and the weight for pose mAP_{50-95} to 0.25.

4.2 Transfer Learning (TL)

Transfer Learning (TL) is an effective technique in machine learning that uses a pre-trained model that is re-purposed and fine-tuned to tackle a related, but different problem by exploiting the knowledge gained from the previous task. TL can be especially useful in domains that lack large-scale datasets necessary to train CNN models from scratch [83]. In addition, when the two

[1]YOLOv7 GitHub: `https://github.com/WongKinYiu/yolov7/tree/pose`
[2]YOLOv8 GitHub: `https://github.com/ultralytics/ultralytics`

tasks are substantially similar, the model can adapt to the new task with minor modifications. It can significantly enhance the performance of a sleeping pose estimation model.

Synthetic datasets, despite their realism and diversity, may lack the complexity and variability present in real-world data, which is crucial for pose estimation, especially in unconventional situations. In the conventional HPE problem, many of the SOTA methods have already achieved good results. Many such methods also provide the weights of their models pre-trained on a diverse and large-scale dataset such as the COCO dataset [46]. It contains more than 200,000 images and 250,000 person instances labeled with keypoints. Using these pre-trained weights allows the HPE model to inherit valuable knowledge and representations about the real world as well as human poses from the real datasets. Representing this size and diversity of real-world knowledge is challenging for synthetic datasets. Generating a large-scale synthetic dataset does not guarantee good generalization on real-world applications. Therefore, using TL techniques can reduce the amount of data required for training and improve the performance of *Sim2Real* transfer.

Since the problem of pose estimation of occluded humans in horizontal positions is a subset of HPE, this process can accelerate the convergence during training and encourage the model to better generalize to previously unseen sleeping poses from the synthetic data. TL, in this context, can help in reducing the gap between synthetic datasets and real-world data. It can enhance the model's capability to learn intricate sleeping poses from synthetic data while still being able to perform well in the real world.

4.3 Evaluation Metric

The evaluation metrics in HPE are described in detail in Sec. 2.3.2. The OKS metric has been very popular in recent years due to its effectiveness in evaluating the accuracy and robustness of HPE methods. The usual Euclidean distance-based metrics vary considerably based on the scale of an object or the type of a keypoint. The OKS metric is inherently scale-invariant and gives more importance to certain keypoint than others [55]. As shown in Fig. 2.5, the facial keypoints have a lower tolerance and hence, a stricter penalty than other keypoints like hips, shoulders, and knees for the same pixel-level prediction error. This means that the metric puts lower tolerance on the keypoints for which a small pixel-level deviation can result in a completely wrong prediction. For other keypoints such as hips or knees, the tolerance level is higher due to the inherent ambiguity about the specific location of these keypoints and the large area where it can be considered correct.

Ronchi *et al.* [73] have effectively visualized it (see Fig. 4.1). The comparison in the figure shows that even if the eye and wrist have the same pixel-level error in their prediction, the OKS value of the wrist is higher (0.75) compared to that of the eye (0.25). This is due to the different tolerance levels of these keypoints. Although the prediction of the eye is incorrect, the prediction of the wrist can still be considered correct. Due to these reasons, the OKS-based pose mAP_{50} and mAP_{50-95} metrics are used to evaluate many recent HPE methods. The same is hence also chosen for this thesis with more focus on the lenient pose mAP_{50}. This is because accurately

determining the locations of the keypoints obscured by the blankets only based on RGB images is a challenging task.

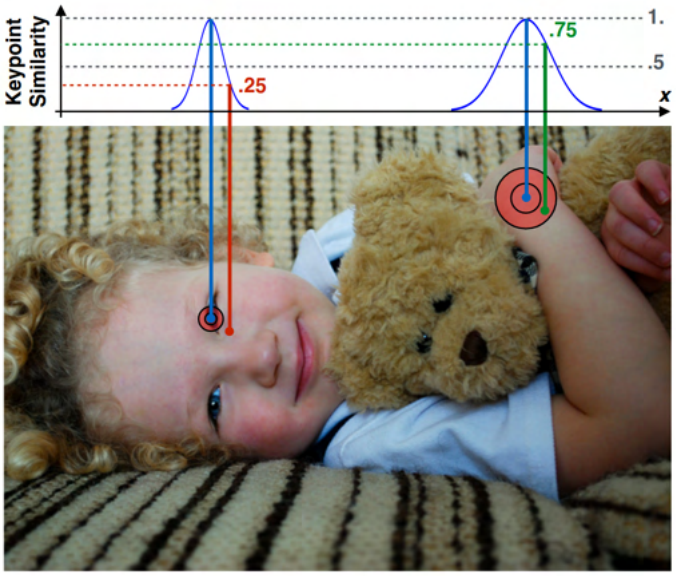


Figure 4.1: OKS between two detections, eye (red) and wrist (green), and their corresponding ground truth (blue). The red concentric circles represent OKS values of 0.5 and 0.85 in the image plane and their size varies by keypoint type. As a result, detections at the same distance from the corresponding ground truth can have different OKS values [73].

4.4 Preparing the Datasets

For facilitating further experiments about the optimal amount of training images needed, multiple datasets with varying amounts of images are prepared. The number of images in the training set varies from 125 to 5000 images. The images in these datasets are randomly selected from the RealPose and RandPose image pools generated earlier. A best practice in training ANNs is to include around 60-80% of all images in the training set, 10-20% in the validation set, and 10-20% in the test set. Following this, the dataset is split into the ratios of 70%, 15%, and 15% of all images for train, val, and test sets respectively. For a good indication of performance, the validation set requires some real images. Hence, the validation set in the training process consists of 50% synthetic data and 50% real data from the SLP dataset. The idea is to evaluate the best-performing model on the mixed validation set and verify the results on a completely real test set. Furthermore, mixed training using some of the real data in the training images is also explored later. The SLP dataset is split into three different parts according to the cover conditions - *uncover*, *cover1* (a thin sheet with $\approx$ 1 mm thickness), and *cover2* (a thick blanket with $\approx$ 3 mm thickness). The combined set of *cover1* and *cover2* is called *cover*. This is done to evaluate the accuracy independently on all the cover conditions of the dataset.

4.5 Domain Adaptation (DA)

As previously discussed, one of the challenges in obtaining real datasets is generating precise annotations. In the context of synthetic data-based learning, unsupervised DA techniques are often used to mitigate this problem. These methods make use of labeled synthetic data and unlabeled real data to enhance the *Sim2Real* transfer. This provides a good alternative for avoiding the expensive annotation process if unlabeled real images can be easily obtained. The method chosen for DA is described below along with the implementation in the YOLOv7 model.

4.5.1 Strong-Weak Distribution Alignment (SWDA)

For bridging the domain gap, some methods try to match the feature distributions of the two domains as illustrated in Fig. 2.12. These methods reduce the difference between the distributions by training a feature extractor. Saito *et al.* [75] introduced Strong-Weak Distribution Alignment (SWDA), a robust approach to unsupervised domain adaptation in object detection. Faster-RCNN [69] is used as a base detector for their method. The SWDA method strongly aligns the local features such as colors and textures of the source images to the target images. Conversely, it performs a weak alignment of the global features of the two domains as strongly matching these distributions can fail due to varying backgrounds and scene layouts across domains. Strong alignment puts a stricter penalty for mismatching feature distributions resulting in a tighter alignment while weak alignment puts a weaker penalty. The local features are extracted from the lower layers while global features are obtained just before the RPN in Faster-RCNN. The basic approach of SWDA is shown in Fig. 4.2.

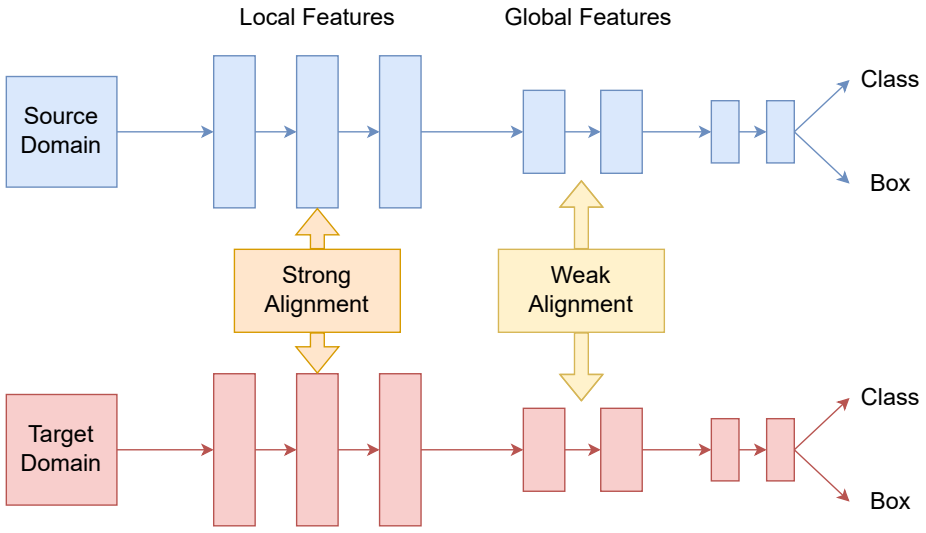


Figure 4.2: Basic idea of the Strong-Weak Distribution Alignment (SWDA) for object detection [75]

For the weak global alignment, the method proposes to utilize a domain classifier that is trained to ignore the easy-to-classify samples and instead focus on the hard-to-classify samples. This achieves a weak alignment of the global features of the domains. The domain label d is set to 1 for the source domain and 0 for the target. It adds a modulating factor $f(p_t)$ in the cross-entropy loss $\log(p_t)$, resulting in

$$-f(p_t)\log(p_t)$$
$$\text{where } p_t = \begin{cases} p & \text{if d=1} \\ 1-p & \text{otherwise} \end{cases}$$

where $p \in [0, 1]$ is the model's estimated probability for the class with label $d = 1$. The loss function chosen is the focal loss [45]:

$$FL(p_t) = -f(p_t)\log(p_t), f(p_t) = (1-p_t)^{\gamma}$$

where γ controls the weight on hard-to-classify examples. The local domain classifier is a simple CNN. The sum of local and global alignment losses for both domains is referred to as adversarial loss. Context vectors are extracted from the intermediate layers of the local and global domain classifiers which contain the information of the whole image. These vectors are concatenated with the features of each region. This is done to improve the performance and stabilize the training of the domain classifiers. The domain classifiers are trained to minimize the detection loss L_{det} as well as the adversarial loss. The overall objective is,

$$\max_{D} \min_{F,R} L_{det} - \lambda L_{adv}$$

where λ controls the trade-off between detection loss and adversarial training loss. The sign of gradients is flipped by a gradient reversal layer.

4.5.2 Integrating SWDA with YOLOv7-pose's architecture

SWDA [75] originally uses Faster-RCNN [69] as a base detector, but the authors claim that it should also apply to other single or two-staged detectors like YOLO [66]. The implementation of SWDA is provided on GitHub by the authors[3]. In this work, the SWDA method is integrated into the model architecture of YOLOv7. For extracting the local features, the 11th layer from YOLOv7's backbone is chosen whereas the 46th layer is chosen for the global features. The 46th layer is the last layer of the backbone of YOLOv7, hence this corresponds to the implementation by the authors where they choose these features just before the RPN. The 11th layer is chosen

[3]SWDA Code: `https://github.com/VisionLearningGroup/DA_Detection`

in preliminary experiments for the local features. The source domain is the synthetic dataset whereas the target domain is the SLP dataset. The domain label is considered as 0 for synthetic images and 1 for real images.

For the real images, a partial forward pass is used where the images only pass through the backbone of the model. The extracted features of the synthetic and real images are passed through a gradient reversal layer and then through the respective domain classifiers which output a domain prediction. For the synthetic images, intermediate context vectors from the domain classifiers are also obtained which are shown in Fig. 4.3 as v_1 and v_2. Four domain classifier outputs (two each for source and target) are then used to calculate the adversarial loss. This loss is added to the normal losses YOLOv7 calculates and is then back-propagated. YOLOv7 has four anchors to calculate the features at multiple scales. The features extracted from these four anchors are passed to the final keypoint detection head. The context vectors for the synthetic images are concatenated to these features. Average pooling and expansion of the tensors are performed on the context vectors to match the dimensions to the feature vectors. This version of YOLOv7 with integrated SWDA model is called **YOLOv7-DA** henceforth.

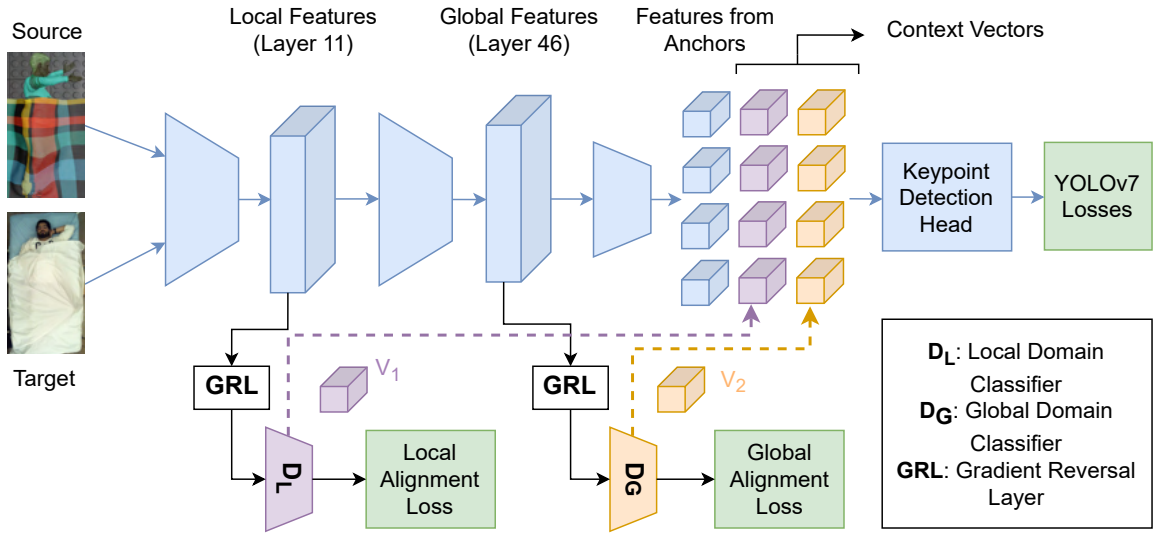


Figure 4.3: Network architecture showing the integration of Strong-Weak Distribution Alignment (SWDA) into YOLOv7-pose

5 Experiments and Results

In the upcoming chapter, the focus is shifted to the experiments with the generated synthetic dataset. It comprises training and evaluation of the model and analysis of results. The performance of each model is compared to analyze their strengths and weaknesses in the context of specific challenges of this thesis. Various training setups are tested to improve the training and to determine the optimal setup for the inclusion of synthetic data in the learning process of the models. Some of these tests are carried out to set a baseline for the performance of the SOTA methods for conventional HPE on the SLP dataset. The two versions of the dataset, namely RealPose and RandPose are also compared. The aim is to determine not only the most effective data generation method but also the most robust model and the best training configuration that will provide a reliable solution for the pose estimation under occlusion. Furthermore, the effect of including varying amounts of real data in the training process is also explored in this section. The results from testing the best models on the SLP dataset are also shown. Furthermore, the results with DA are also explained.

5.1 Performance of Pre-trained Models on the SLP Dataset

To set a baseline, the performance of the pre-trained YOLOv7 and YOLOv8 models was evaluated on the three subsets of the SLP dataset. YOLOv7 performed well on the *uncover* set of the SLP dataset. While on the *cover1* and *cover2* set, the performance dropped drastically. The same was also observed for YOLOv8. These performances are summarized in the Table 5.1 and 5.2. Some examples of inference by YOLOv7 on uncovered and covered images are shown in the Appendix, Fig. 1, 2, and 3.

Table 5.1: Performance of the pre-trained YOLOv7 on the SLP dataset

SLP subset	Box mAP_{50}	Box $mAP_{(50-95)}$	Pose mAP_{50}	Pose $mAP_{(50-95)}$
uncover	1	0.993	1	0.882
cover1	0.00145	0.000264	0.361	0.0561
cover2	0.00444	0.00286	0.348	0.0546
cover	0.00134	0.000296	0.351	0.0585

Table 5.2: Performance of the pre-trained YOLOv8 on the SLP dataset

SLP subset	Box mAP_{50}	Box $mAP_{(50-95)}$	Pose mAP_{50}	Pose $mAP_{(50-95)}$
uncover	0.995	0.995	0.995	0.891
cover1	0.0365	0.00848	0.424	0.0732
cover2	0.0466	0.0106	0.422	0.0728
cover	0.0408	0.00992	0.422	0.0731

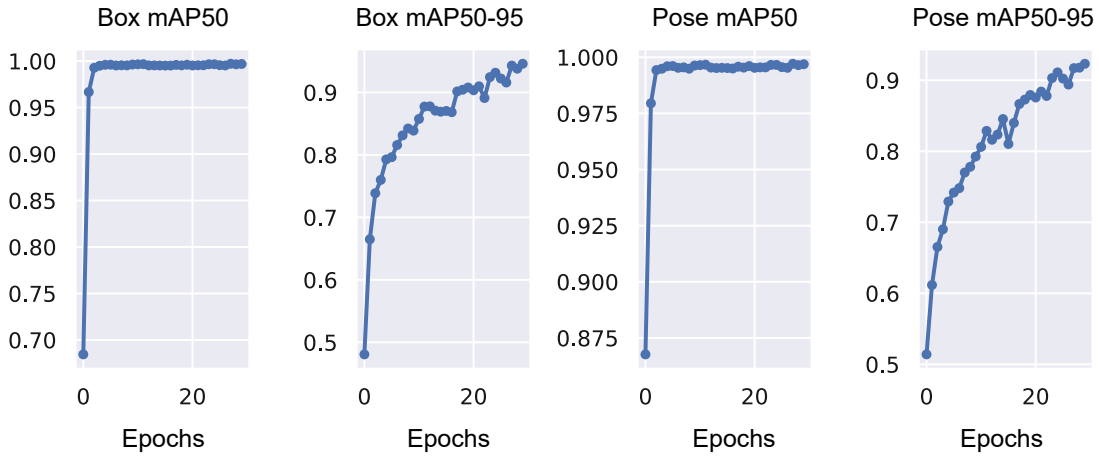
This comparison suggests that the pre-trained models perform well when the person is completely visible in the overhead image. They can predict the extreme, unconventional sleeping poses represented in the SLP dataset proving the quality of these methods. However, even the accuracy of the bounding boxes in the *cover1* and *cover2* subsets suffers notably indicating that some retraining has to be done to be able to use these models on the SLP dataset. According to the Maji *et al.* [55], the method on which YOLOv7 is based should be able to detect keypoints even outside the bounding box. Hence the poor performance of the box as well as pose metrics further suggests a need for retraining. It is important to note, that the pose estimation metrics seen here are for all 17 keypoints, out of which 5 to 7 keypoints (5 facial keypoints and 2 for shoulders) are always uncovered in the images. Hence, these pre-trained models are still able to produce the scores above.

5.2 Training Process of the Models

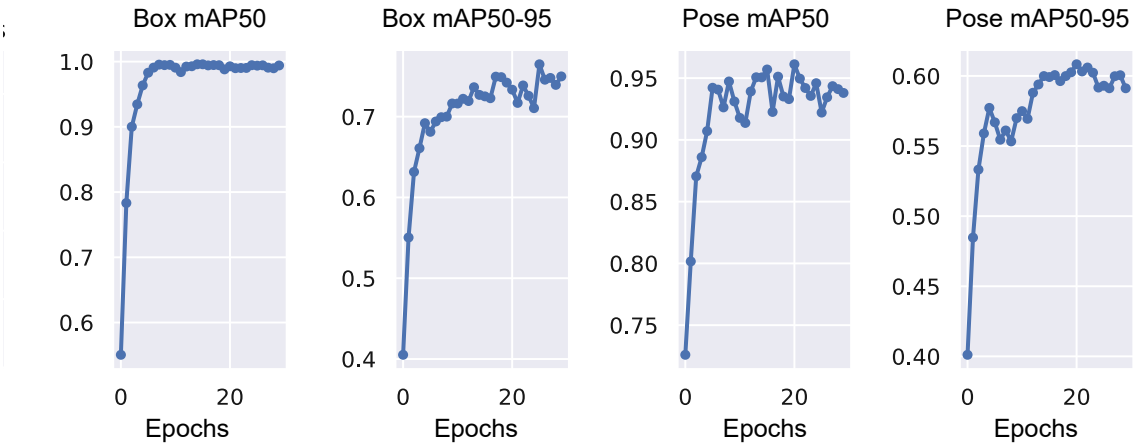
For comparison of the YOLOv7 and YOLOv8's training process with the default hyperparameters provided by the authors, both of these pre-trained models were retrained separately on the SLP dataset as well as the synthetic dataset. Due to its simplicity, the RealPose dataset was chosen as the base synthetic dataset for this comparison. The train, val, and test sets contained images from the same domain to have a fair comparison. The models were trained on 500 images in the default configuration for 30 epochs. When training YOLOv7, the pose and box metrics were showing stable and continuous increment, both with the real as well as synthetic data as seen in Fig. 5.1.

However, some irregularities were seen in training the YOLOv8 model. When training with the SLP dataset, the box and pose losses during training increased for a few epochs. The pose mAP_{50} and mAP_{50-95} metrics also saw a slight decline before starting to increase again (see Fig. 5.2a). This behavior was aggravated on the synthetic dataset with a larger drop (see Fig. 5.2b). The metrics started increasing again after the initial decline, but almost 20 epochs were required before they returned to their initial value. Many different configurations with varying learning rate, warmup epochs, and different weights for the box and pose losses were tried to solve this issue, but none of them proved to be effective. Another pre-trained model offered by YOLOv8, namely *yolov8x-pose* was also tried with the synthetic data and still, the same behavior was observed (see Fig. 5.2c). The training curves were still not stable and frequent oscillations

and jumps were observed. This indicates some instability in training which can affect the model's performance. The model is unable to learn relevant features to decrease the pose and box losses. This might cause the model to learn unwanted features from the images for these initial epochs. The features learned this way might not be unlearned with additional training. However, the behavior of the box and pose mAP_{50} metrics was slightly better than *yolov8x-pose-p6* with less frequent and less intense oscillations. Hence, in the next parts of this thesis, *yolov8x-pose* was chosen as the pre-trained model and is referred to as YOLOv8.

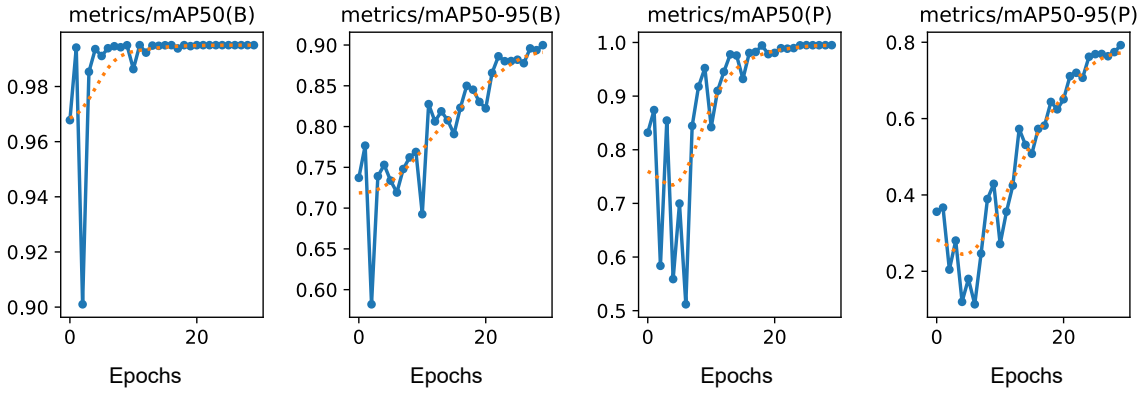


(a) With 500 images from the SLP dataset

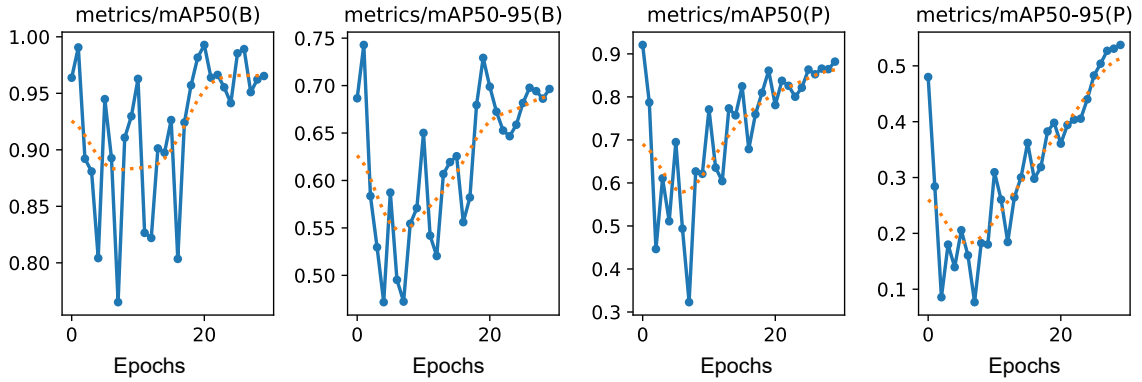


(b) With 500 images from the RealPose dataset

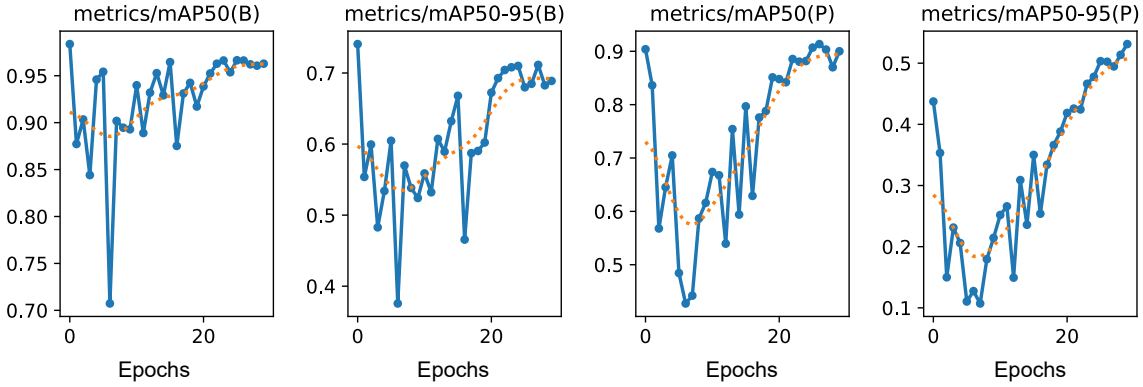
Figure 5.1: Training process of pre-trained **YOLOv7**



(a) With 500 images from the SLP dataset (Pre-trained weights from **yolov8x-pose-p6**)



(b) With 500 images from the RealPose dataset (Pre-trained weights from **yolov8x-pose-p6**)



(c) With 500 images from the RealPose dataset (Pre-trained weights from **yolov8x-pose**)

Figure 5.2: Training process of pre-trained **YOLOv8**. *B* indicates box metrics and *P* indicates pose metrics.

5.3 Optimizing the Transfer Learning Configuration

This section deals with finding the optimal training configuration for utilizing transfer learning with the chosen HPE models and the generated datasets. Experiments regarding the amount of training data and freezing different numbers of layers in the models are described below.

5.3.1 Epochs and Amount of Training Data

To determine the optimal amount of data required for training, several training runs were conducted using different sizes of training datasets. The number of images in the training sets was 125, 500, 1000, 2000, 3000, and 5000. The comparison graphs for the pose estimation metrics can be seen in Fig. 5.3. These experiments show that the best-performing model on the validation set was not necessarily the one with the highest number of training images or more epochs. In this experiment, the best-performing model for YOLOv7 with 125 training images was at the 47th epoch, with 250 images was at the 36th epoch, with 500 images was at the 16th epoch, and with 1000 images was at the 3rdepoch. As mentioned earlier, this epoch was determined by a combination of pose mAP_{50} and mAP_{50-95}. As expected, with increasing the training data, the model reached its optimal performance earlier in the training process. This indicates that only a small amount of training data is needed for optimal performance of the model. However, the models trained with more than 1000 images show a performance drop even with a smaller number of epochs. For example, the pose mAP_{50} and mAP_{50-95} metrics see a sharp decline in the initial epochs as seen in Fig. 5.3. While these metrics improve after a few epochs, they don't reach the same accuracy set by the models trained on fewer images. Also, this increase in performance in the later epochs can be attributed to the model overfitting on the synthetic data while performing poorly on the real data in the validation set. To verify whether the best-performing models on the validation set also maintain the accuracy on the real data, these models were also tested on the *cover1* and *cover2* subsets of the SLP dataset. The models trained on 250-1000 images consistently showed a mAP_{50} above 0.7 and a mAP_{50-95} above 0.19 on *cover1* and *cover2*. The models trained on less than 250 and more than 1000 images did not reach this accuracy. Hence, it can be concluded that training with more than 1000 images degrades YOLOv7's performance. Only the training sets of up to 1000 images are considered in the later parts of this thesis.

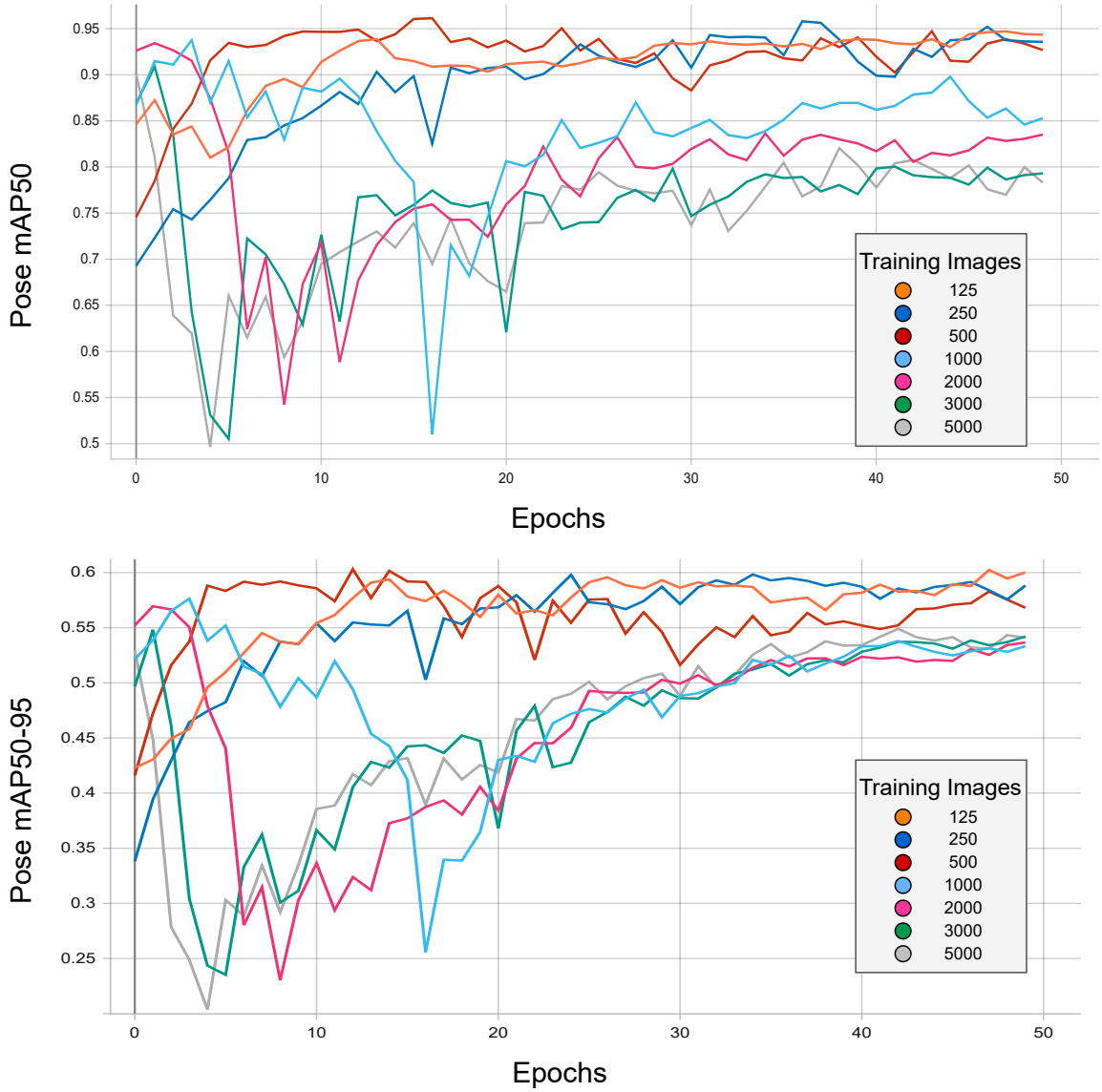


Figure 5.3: Pose mAP_{50} and mAP_{50-95} graphs showing comparison for training YOLOv7 with varying amount of data for 50 epochs

In the case of YOLOv8, the training process is not as stable. The pose metrics saw a sharp decline while training and took more than 20-30 epochs to recover and reach their initial value as also seen in Fig. 5.2. The behavior did not change with a varying number of training images, learning rates, or epochs. On the validation set, the best pose accuracy was achieved with a higher amount of training images. Similarly to YOLOv7, the best-performing models on the validation set were also tested on the *cover1* and *cover2* sets of the SLP dataset. The pose mAP_{50} and mAP_{50-95} metrics consistently dropped with increasing the number of training images. The model's higher accuracy on the mixed validation set may be due to its good performance on

synthetic images but poor performance on real images in the set. This suggests that the model is not able to transfer the knowledge learned on the synthetic dataset to the real dataset. This is a disadvantage for YOLOv8 as compared to YOLOv7 in the scope of this thesis.

The initial image pools generated for both datasets contained more than 10000 images. The synthetic image generation process is computationally expensive due to the realism of the characters, the presence of a large number of textures, and most importantly, the soft-body physical simulation of the blanket colliding with the character. All these elements were essential for a high-quality synthetic dataset with an accurate representation of the blanket occlusion. Hence, a lesser amount of data needed for optimal training is always desired. Less amount of synthetic data used for retraining also means that the models will learn only what is necessary from the synthetic domain and retain most of the knowledge from the real domain. This can help the model generalize better on real-world images. With a higher amount of training data, the model may start to overfit on the synthetic data and hence, lose its domain knowledge and accuracy on the real images.

5.3.2 Freezing of Layers

In the context of TL, freezing layers refers to freezing or fixing the weights of some layers during the training process. This is intended to retain the knowledge from the pre-trained models to expedite the training process and improve generalization. It also reduces the risk of overfitting these layers to the new data [78]. For fine-tuning a network on new data, the shallow layers of the networks are usually frozen with the deeper layers getting retrained. Shallow layers are responsible for learning low-level features, like edges, corners, colors, and contours. Deeper layers typically learn more high-level characteristics containing more complicated details about the image learned from low-level feature pairings [64]. These may include things like faces, limbs, and other body parts in case of HPE.

YOLOv7 has 118 layers before the final keypoint estimation head. Out of these layers, the first 46 layers are from the backbone of the model while the rest are from the head of the model. For fine-tuning these models on the synthetic dataset, multiple experiments were conducted by freezing varying numbers of the shallower layers. For YOLOv7, the first 46, 63, 74, and 91 layers in the model were frozen. The best-performing models were chosen in the same manner as before - based on a combination of pose mAP_{50} and mAP_{50-95} values on the validation set. Training runs were conducted with sets of 125, 250, 500, and 1000 training images. Fig. 5.4 shows an example of the pose mAP_{50} and mAP_{50-95} graphs on validation data while training YOLOv7 with 250 training images. The best-performing YOLOv7 model is always the one with no frozen layers, even when training with varying amounts of training images. Thus, it was observed that freezing the layers of the model hurt its performance. This result might be because the shallower layers could not adjust to the new local features resulting from blanket-based occlusion. Therefore, in all future experiments, no layers are frozen during the training process.

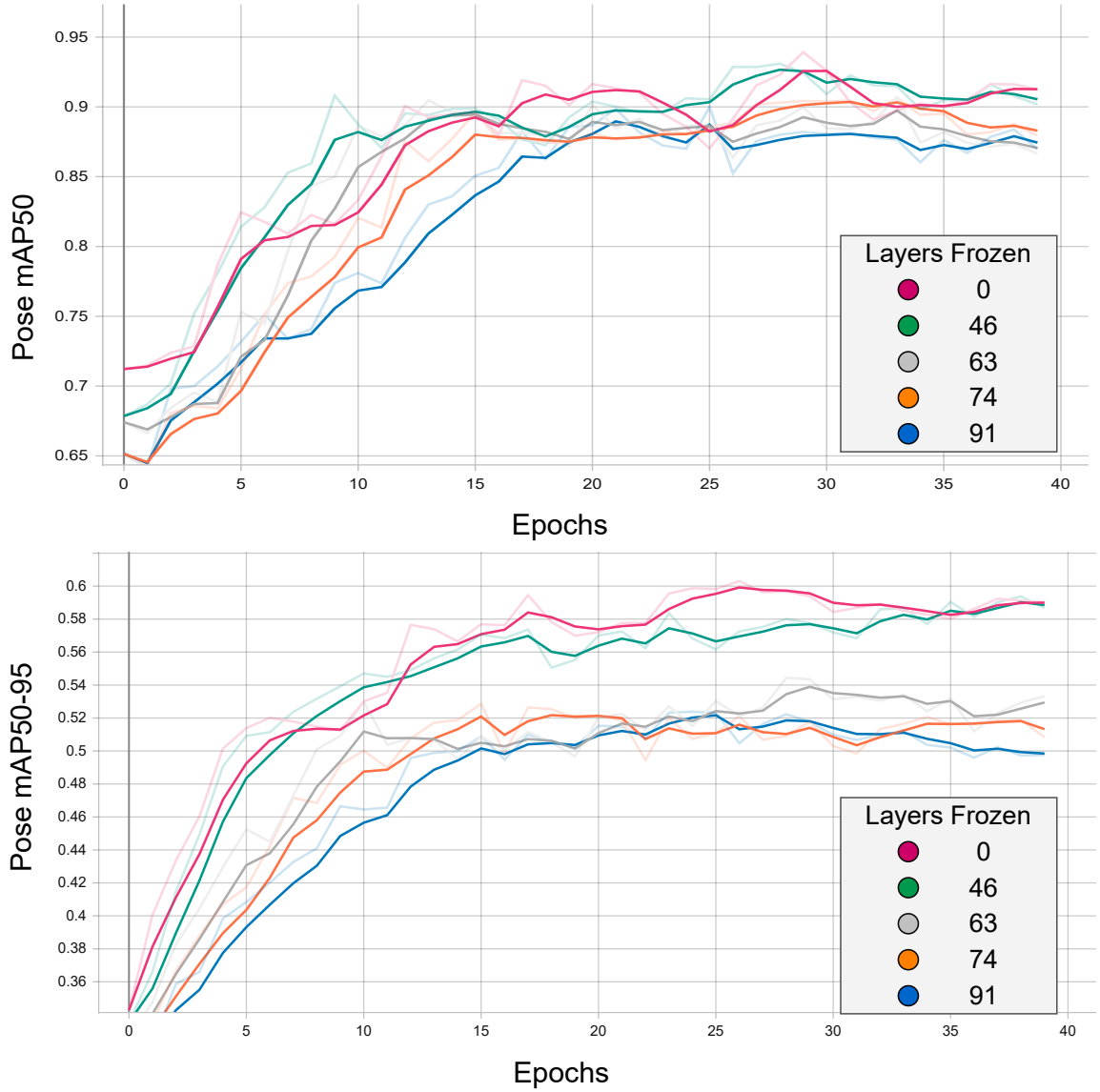


Figure 5.4: Pose mAP_{50} (top) and mAP_{50-95} (bottom) graphs showing accuracy with freezing different number of layers while training YOLOv7 on 250 images. The same graphs for 500 images are shown in the Appendix, Fig. 8. (Graphs are smoothed using a 0.5 factor for better visualization; non-smoothed graphs appear translucent.)

Similarly, the YOLOv8 model comprised 28 layers with the first 12 layers as the backbone and the remaining layers forming the head of the network. Many of these 28 layers represented layer groups instead of individual layers in the case of YOLOv7. Following YOLOv8's numbering convention, the first 12, 18, 21, and 24 layers were frozen. The rest of the training setup was the same as in YOLOv7. In contrast to YOLOv7, the best-performing models on the validation set were the models with a higher number of frozen layers. However, as also seen in Sec. 5.3.1, this

accuracy is inconsistent with the SLP dataset. Even for the best-performing models, the pose metrics dropped by around 20% points on the SLP dataset. Unlike in YOLOv7, the bounding box metrics also saw a decline. Many images had no bounding box detection and hence, no keypoint detection. The retrained model also failed to accurately detect the occlusion-free keypoints in the SLP dataset. A few examples showing these problems with YOLOv8 are shown in the Appendix, Fig. 9. Due to these factors, the YOLOv8 is not considered for the thesis, and in the next sections, only the experiments and results from YOLOv7 are presented.

5.4 Performance Comparison of RealPose and RandPose Dataset

For comparing the two versions of datasets, YOLOv7 was used with the optimized configuration for transfer learning determined in Sec. 5.3 - 500 training images with no frozen layers. The training was run for 30 epochs. The best-performing model for both datasets was evaluated on the three subsets of the SLP dataset. The best metrics obtained for RealPose and RandPose datasets are summarized in Table 5.3. The initial assumption was that the RandPose dataset would get better accuracy on the SLP dataset due to the higher pose diversity in the data. However, the initial pose metrics obtained with RandPose were around 4 to 8% points lower than with the RealPose dataset. To try to boost the accuracy of the model retrained with the RandPose dataset, many trials were performed, namely training with a validation set consisting of 100% real images, training with different learning rates (learning rate), training with different learning rates for backbone and head layers. Lower learning rate for all the layers was the most successful which resulted in the accuracy reported in Table 5.3. After these experiments, the model trained with the RealPose dataset still outperformed the model trained with the RandPose dataset in the pose mAP_{50-95} metric. However, for the pose mAP_{50} metric, the RandPose dataset resulted in better performance, especially in the *cover2* set. Therefore, for purely synthetic data-based training, both datasets are effective in certain situations but there is a slight edge for the RealPose data. This may be due to the more realistic pose representation in the RealPose dataset or the problems with blanket simulation in the RandPose dataset described in Sec. 3.3.3.

Table 5.3: Performance comparison of YOLOv7 on the SLP dataset after retraining with **RealPose** and **RandPose** dataset. Values in **bold** indicate the better-performing dataset for the particular metric.

	RealPose		**RandPose**	
SLP set	**Pose mAP_{50}**	**Pose mAP_{50-95}**	**Pose mAP_{50}**	**Pose mAP_{50-95}**
cover1	**0.743**	**0.212**	0.732	0.197
cover2	0.672	**0.197**	**0.696**	0.176
cover	0.708	**0.205**	**0.711**	0.185

Table 5.3 also serves as a measure of the performance boost that was obtained by retraining

YOLOv7 with only synthetic data. With the datasets generated in this work, the pose mAP_{50} score was increased by approximately 37% points for the *cover1* set and approximately 32% points for the *cover2* set. The pose mAP_{50-95} scores which were almost negligible for the pre-trained model, also see a considerable increase after retraining. This demonstrates the ability of synthetic data to teach the model the pose representation of humans covered with a blanket. However, the model's precision in estimating occluded keypoints still needs improvement. Some qualitative result examples can be seen in Appendix, Fig. 10. As a result, further experiments were conducted to enhance the accuracy of the model in the next sections.

5.5 Varying the Lower Body Sigma Values in the OKS Loss

The main challenge of this thesis is to accurately predict the occluded lower body keypoints. Due to the unique nature of this problem and the OKS-based loss function used in YOLOv7, a hypothesis was formed regarding the per-keypoint standard deviation values. The OKS loss function in YOLOv7 by default used the σ values as mentioned in Fig. 2.5. As explained in Sec. 4.3, these values control the tolerance of each keypoint and hence, also its weight in the OKS loss. As a result, setting a lower σ value and hence a stricter penalty for a particular keypoint should make the model focus more on accurately predicting the keypoint. Since for the task of this thesis, the model should focus more on the occluded keypoints of the lower body, multiple training runs were conducted with an identical setup but with varying lower body sigma values. The keypoints considered as lower body keypoints were the hips, knees, and ankles. The sigma values for these keypoints were divided by a factor f such that

$$\sigma_{lowerbody,new} = \frac{\sigma_{lowerbody,original}}{f}$$

YOLOv7 was trained with the RealPose dataset using the same training configuration as described in Sec. 5.4. For the hypothesis to be verified, gradually increasing the factor f should result in a steady increase in accuracy. However, as seen from Table 5.4, no definite trend in the accuracy was observed. The best performance was still achieved with the default σ values. Hence, this hypothesis could not be proved with the experiment, and the default values of σ are used for all further experiments.

Table 5.4: Performance of YOLOv7 on the SLP dataset after retraining with **RealPose** with varying σ values in the OKS loss.

	Pose mAP_{50}								
SLP set	$f = 1$	$f = 1.2$	$f = 1.4$	$f = 1.6$	$f = 1.8$	$f = 2$	$f = 3$	$f = 4$	$f = 5$
cover	**0.708**	0.624	0.676	0.7	0.633	0.68	0.645	0.633	0.648

5.6 Mixed Training with Real Data

The performance of the retrained YOLOv7 model on the SLP dataset suggests modest pose mAP_{50} values but unsatisfactory pose mAP_{50-95} values (see Table 5.3). To investigate if the performance could be further improved by adding small amounts of real data, samples from the *cover1* and *cover2* sets of the SLP dataset were mixed in the training set of the model. The training set comprised a total of 500 training images including synthetic as well as real data. The pre-trained YOLOv7 was trained on these images for 30 epochs and the best-performing model was then tested on the SLP dataset. The percentage of real data was at first set to 10% of the total training images. This was gradually reduced to examine the performance boosts that can be obtained with as little real data as possible.

The results of mixed training with real data can be seen in Table 5.5 and 5.6. As expected, a higher amount of real data included in the training was accompanied by an increase in accuracy. Additionally, it can be observed that with the same amount of real data, the model trained on the RandPose dataset consistently outperforms the model trained on the RealPose dataset. This observation is in contrast to the one in the Sec. 5.4 where the model trained with the RealPose dataset performed marginally better than the models trained with the RandPose dataset.

Table 5.5: Performance of YOLOv7 on the SLP dataset after mixed training with the **RealPose** dataset and a varying amount of real data.

Pose mAP_{50}					
SLP set	**0% Real data**	**1% Real data** (5 Images)	**2.5% Real data** (12 Images)	**5% Real data** (25 Images)	**10% Real data** (50 Images)
cover1	0.743	0.79	0.883	0.924	0.974
cover2	0.672	0.738	0.831	0.872	0.927
cover	0.708	0.764	0.858	0.899	0.941

Pose mAP_{50-95}					
SLP set	**0% Real data**	**1% Real data** (5 Images)	**2.5% Real data** (12 Images)	**5% Real data** (25 Images)	**10% Real data** (50 Images)
cover1	0.212	0.264	0.333	0.394	0.479
cover2	0.197	0.241	0.305	0.356	0.448
cover	0.205	0.251	0.318	0.375	0.463

Table 5.6: Performance of YOLOv7 on the SLP dataset after mixed training with the **RandPose** dataset and a varying amount of real data.

Pose mAP_{50}					
SLP set	**0% Real data**	**1% Real data** (5 Images)	**2.5% Real data** (12 Images)	**5% Real data** (25 Images)	**10% Real data** (50 Images)
cover1	0.732	0.865	0.913	0.948	0.974
cover2	0.696	0.762	0.843	0.893	0.939
cover	0.711	0.811	0.873	0.921	0.956

Pose mAP_{50-95}					
SLP set	**0% Real data**	**1% Real data** (5 Images)	**2.5% Real data** (12 Images)	**5% Real data** (25 Images)	**10% Real data** (50 Images)
cover1	0.197	0.293	0.343	0.415	0.519
cover2	0.176	0.259	0.314	0.367	0.48
cover	0.185	0.275	0.325	0.391	0.499

Including real images in the dataset helps to decrease the gap between different domains, which leads to an increase in accuracy. This happens because the model learns domain-specific knowledge from real images. Therefore, the results suggest that having knowledge of not only the real-world domain but also the dataset domain is crucial for achieving better performance from the model. This "dataset" domain refers to the feature representation for the SLP dataset in particular. Due to the lack of diversity in the SLP dataset as explained in Sec. 3.2, the feature distribution of the dataset's domain can be confined and needs to be learned for a good performance on the dataset. The issues faced during the blanket simulation while creating the RandPose dataset, as described in Section 3.3.3, might have resulted in the poorer performance of the model trained completely on the RandPose dataset. The observation also hints towards a higher domain gap in the RandPose dataset, which prevents effective generalization to the SLP dataset when training solely on synthetic data. However, including a few real images helps bridge this gap, and the higher pose diversity represented in the RandPose dataset can be effectively learned and transferred to the SLP dataset. It is important to note that good performance can be achieved by including a small amount of real data along with both the RealPose and RandPose datasets.

Comparing the 1% real data with 10% real data in Table 5.6, it can be seen that the pose mAP_{50} value for *cover1* increased by 10.9% points whereas the same for *cover2* increased by 17.7% points. The pose mAP_{50-95} values on the other hand, increased by 22.6% points for *cover1* and 22.1% points for *cover2*. The steeper rise in the stricter pose mAP_{50-95} metric suggests that the real data plays an important role in enabling the precise location of the keypoints. An approximate pose of the lower body can also be estimated or "guessed intelligently" based on the contours created by the blankets and the upper body keypoints that remain unobstructed in most

of the images. This knowledge can be learned from the synthetic data, but for a higher precision of the completely occluded keypoints, the importance of real data cannot be denied.

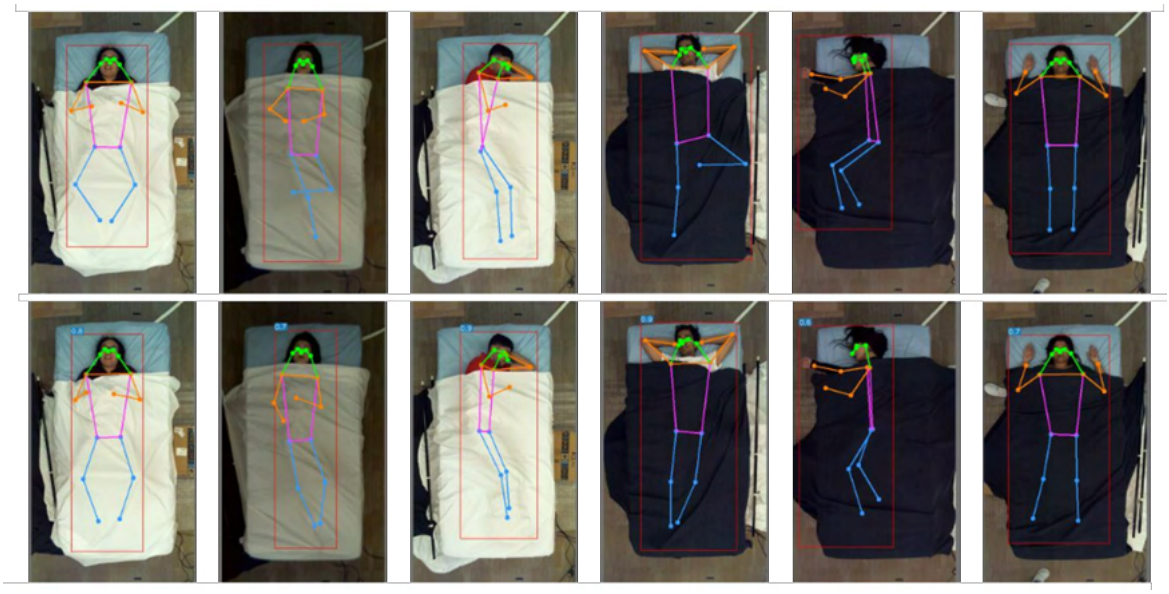


Figure 5.5: Qualitative results of mixed training with **RandPose** and **2.5**% real data shown with ground truth (top) and predictions (bottom)

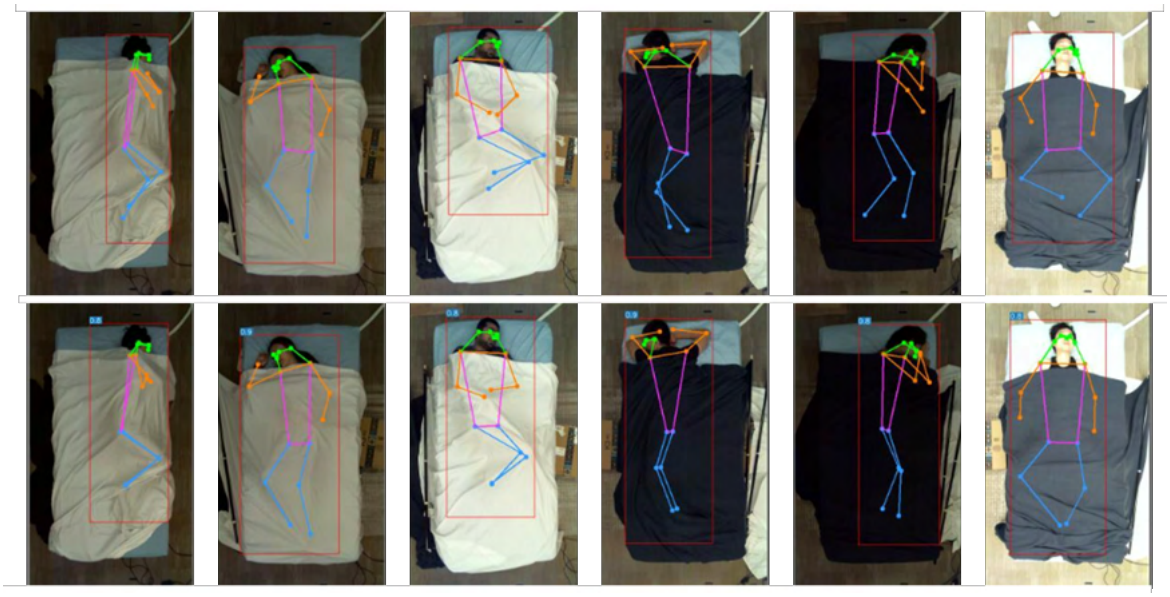


Figure 5.6: Qualitative results of mixed training with **RandPose** and **10**% real data shown with ground truth (top) and predictions (bottom)

Some examples from the results of mixed training can be seen in Fig. 5.5 for 2.5% real data and

Fig. 5.6 for 10% real data. Fig. 5.5 illustrates that the model is capable of predicting the pose with considerable accuracy, even when dealing with severe occlusions. The model may fail to accurately predict knee and ankle keypoints in extreme poses with significant bends in the thigh or knee. This drawback of the model is significantly reduced by adding more real data to the training process. As seen from the examples in Fig. 5.6, the model successfully predicts even the extreme poses with good accuracy.

The results obtained in this section demonstrate the effectiveness of synthetic datasets in decreasing the dependency on real data. However, it also emphasizes the importance of real data in *Sim2Real* transfer. In real-world scenarios, the expensive process of generating a large amount of annotated real data can be significantly reduced with a combination of synthetic datasets and transfer learning techniques. Synthetic data can also be crucial in situations where generating manual or automatic annotations is challenging. Additionally, the data for rare, extreme, or unsafe situations can be effectively represented in the synthetic dataset, while the real dataset can be simplified to only serve the purpose of bridging the domain gap.

5.7 Training YOLOv7 with Domain Adaptation

It is worth investigating the effectiveness of unsupervised DA methods on the *Sim2Real* transfer from the generated synthetic dataset to the SLP dataset. Such methods can utilize unlabeled real data to bridge the domain gap as explained in Sec. 4.5.1. The implementation of a DA method, SWDA [75] with YOLOv7 as described in Sec. 4.5.2 was performed with 500 training images. With these experiments, the aim was to improve accuracy over the best models in Table 5.3. For the RandPose dataset, the training was done with the same learning rate that resulted in the best accuracy. Different weights for the domain alignment loss have been tried that would optimize the training of the network. The weight for domain alignment loss was initially increased by 1 while some weights smaller than 1 were also tried. Later, to find the optimal weight, training with intermediate weights was performed. The results of this experiment are presented in the Appendix, Fig. 11. The YOLOv7 model retrained with the DA method is referred to as YOLOv7-DA.

Table 5.7 and 5.8 summarize the results of the best-performing YOLOv7-DA models trained with the RealPose and RandPose datasets. The SWDA method yielded better results using the RealPose dataset with improvements of around 3-4% points in the pose mAP_{50} metrics. However, a consistent improvement could not be observed using the RandPose dataset. Only the *cover1* set of the SLP dataset saw improvements in both metrics whereas *cover2* saw a slight decline in accuracy. These initial experiments show that the domain gap can be reduced by utilizing unlabeled real images with the help of such DA methods resulting in better generalization on the real datasets. The implementation of the SWDA method in this work indicates that it can be integrated in a variety of pose estimation methods and possibly also for other CV tasks.

Table 5.7: Best performances of YOLOv7-DA on the SLP dataset after training with **RealPose** dataset (Weight for domain alignment loss = 3.5). Values in **bold** indicate improvement over training without SWDA.

	Without SWDA		With SWDA	
SLP set	**Pose** mAP_{50}	**Pose** mAP_{50-95}	**Pose** mAP_{50}	**Pose** mAP_{50-95}
cover1	0.743	0.212	**0.774**	**0.222**
cover2	0.672	0.197	**0.703**	0.179
cover	0.708	0.205	**0.732**	0.196

Table 5.8: Best performances of YOLOv7-DA on the SLP dataset after training with **RandPose** dataset (Weight for domain alignment loss = 3). Values in **bold** indicate improvement over training without SWDA.

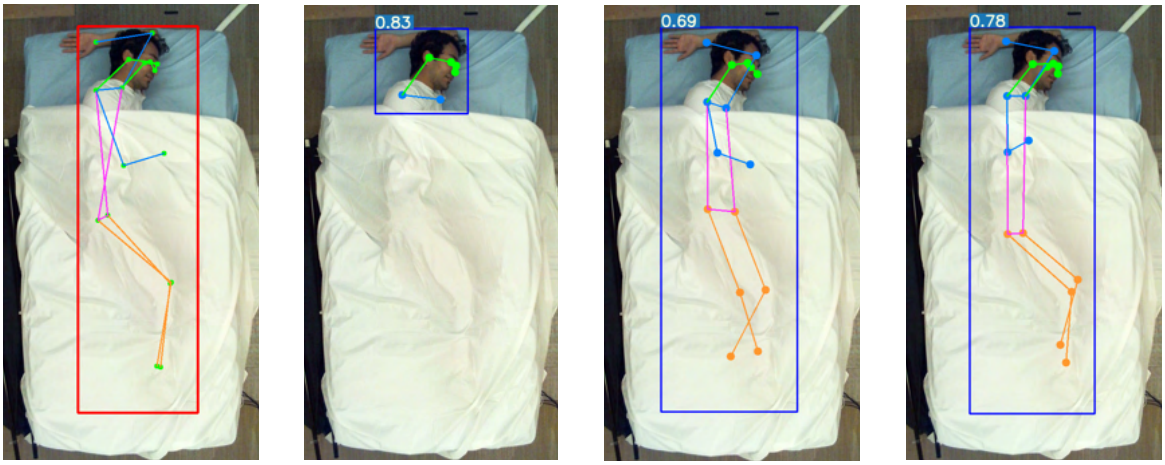
	Without SWDA		With SWDA	
SLP set	**Pose** mAP_{50}	**Pose** mAP_{50-95}	**Pose** mAP_{50}	**Pose** mAP_{50-95}
cover1	0.732	0.197	**0.763**	**0.21**
cover2	0.696	0.176	0.669	0.164
cover	0.711	0.185	**0.714**	**0.186**

5.8 Results Summary

This section provides a summary of the results obtained from the experiments of the thesis. The results shown in Table 5.9 are summed up from Tables 5.1, 5.3, 5.5, 5.6, 5.7, and 5.8. As evident from the table, the pre-trained HPE models perform poorly. The limitations of the models in accurately detecting the pose of humans in horizontal positions under occlusion could be mitigated by leveraging the advantages offered by synthetic datasets. The models retrained only with the synthetic data generated in this work significantly outperformed the pre-trained models. However, the accuracy achieved did not compare to the accuracy of the SOTA pose estimation methods on conventional pose estimation datasets. This indicated room for improvement which was fulfilled by a small amount of real data. Real datasets played an important part in the training process by bridging the domain gap and hence, improving generalization and performance in real-world applications. This was evident by the steeply rising accuracy obtained by adding a higher amount of real data. Unsupervised domain adaptation techniques also helped in bridging the domain gap by using only unlabeled real images. Preliminary experiments conducted in this work with the SWDA method [75] resulted in a decent boost in accuracy. As a result, the time and cost-intensive annotation process of the real images can be avoided.

Table 5.9: Results summary: Performance of YOLOv7 on the SLP dataset after training with the generated synthetic datasets.

Pose mAP_{50}					
RealPose					
SLP set	**Pre-trained**	**100% Synthetic**	**2.5% Real**	**10% Real**	**Domain Adaptation**
cover	0.351	0.708	0.858	0.941	0.732
RandPose					
SLP set	**Pre-trained**	**100% Synthetic**	**2.5% Real**	**10% Real**	**Domain Adaptation**
cover	0.351	0.711	0.873	0.956	0.714



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

Figure 5.7: Qualitative result example from the experiments performed in the thesis. The figure shows ground truth (a), the predictions from the pre-trained YOLOv7 (b), from YOLOv7 retrained with synthetic data and with domain adaptation (c), and YOLOv7 with mixed training using 10% real data (d).

Fig. 5.7 shows an example of the qualitative results obtained. As seen from Fig. 5.7b, the pre-trained models are completely incapable of predicting the joints under a heavy occlusion even from a conventional, overhead viewpoint. This problem was substantially solved by retraining the models with only synthetic data and applying domain adaptation techniques. However, the precision of the estimated joint location was not high. This was evident in the estimation of the

knees and ankles of the human (see Fig. 5.7c). It is worth noting that synthetic data has shown significant potential, despite the lack of visual data about the occluded joints in the SLP dataset (see Appendix, Fig. 4). To achieve precision in estimating heavily occluded joints, a few real images were included in the training data. This allowed the model to learn the feature distribution of the real-world data, resulting in an accurate estimation of the horizontal pose even under heavy occlusions (see Fig. 5.7d).

6 Conclusion and Future Work

This thesis aimed to train deep learning models with synthetic data to accurately estimate human poses in horizontal positions under heavy occlusion using only RGB images. Obtaining real datasets for this task is challenging due to the difficulty, time requirement, cost of manual annotations, and privacy concerns. For this purpose, synthetic data was utilized. The synthetic data generation pipeline developed in previous work was used where the occlusion was introduced with a soft-body simulation of a blanket. In this thesis, the synthetic dataset was modified and optimized for the training of deep learning models. Some state-of-the-art models for human pose estimation were trained with the generated synthetic dataset. Training with only synthetic data resulted in a considerable increase in accuracy, demonstrating the effectiveness of the modifications undertaken in this thesis. Further experiments with including a small amount of real data along with synthetic data in the training process also boosted the accuracy significantly. Finally, an unsupervised domain adaptation method was incorporated into the architecture of YOLOv7-pose which improved the performance by reducing the domain gap with the help of unlabeled real images.

For adapting the synthetic data for training CNN models, the factors enhancing the quality of the synthetic dataset were identified. Two pose generation techniques were used in this work to represent human pose extremities in horizontal positions which resulted in two versions of the dataset, namely RealPose and RandPose. The RealPose dataset contained more realistic human poses while the RandPose dataset contained more diverse and extreme poses. To reduce the domain gap to the real data, several domain randomization techniques were applied, resulting in better generalization of the models trained with the synthetic data on the real data. These modifications increased the effectiveness of the synthetic data for the task of this thesis.

The quality of the synthetic data was indirectly assessed by observing the performance of the retrained models on a real dataset. Here, the Simultaneously-Collected Multimodal Lying Pose (SLP) dataset was used as the real dataset. The recently released architectures, namely YOLOv7-pose and YOLOv8-pose were used for the training process using the transfer learning approach. The task of pose estimation of sleeping humans under occlusion is a subset of conventional human pose estimation. Hence, the previously learned knowledge about in-the-wild human poses from the pre-trained models provided by the YOLO models could be repurposed for the task of this thesis.

To establish a baseline for this research, the pre-trained YOLO models were tested on the SLP dataset. The models performed very well on uncovered humans even with complex sleeping poses. However, a serious drop in performance was observed in the covered humans due to the lack of visibility of the joints and unconventional under-the-blanket pose representation. The aim was to teach the models this knowledge with the synthetic data and improve the performance on

the SLP dataset. Due to the higher diversity of poses represented in the RandPose dataset, it was expected to be more suitable for the *Sim2Real* transfer. Contrary to expectations, the model trained with the RealPose dataset performed marginally better on the SLP dataset as compared to the model trained with the RandPose dataset when training with only synthetic data. This might have been caused by the problems in the blanket simulation encountered during the data generation process of the RandPose dataset.

Training with synthetic data significantly increased the pose estimation metrics of the model. However, the precision in estimating the keypoints was still not very high. This indicates a domain gap in the synthetic datasets. Hence, the next experiments focused on reducing the domain gap using two methods, namely mixed training with real data, and unsupervised domain adaptation. In mixed training with real data, a few real images from the SLP dataset were added to the training set of the model. Experiments were performed to observe the improvement in performance with as few real images as possible. The model trained on RandPose dataset consistently outperformed the one trained on RealPose dataset. This contradicted the earlier conclusion and simultaneously validated the initial hypothesis regarding the RandPose dataset.

The dataset used for the pre-trained models was the COCO dataset [46], which contains over 200,000 images and 250,000 instances of people. The SLP dataset contains more than 10,000 images. Interestingly, a combination of about 50 real images with synthetic data produced good results on the real data. These results demonstrate that even a small amount of real data can significantly improve the model's performance by providing valuable insights about real-world data. Thus, using transfer learning in combination with small-scale synthetic datasets can substantially reduce the dependence on large-scale real datasets. The domain adaptation method also showed potential in bridging the domain gap. The initial experiments with the method resulted in a slight increase in performance compared to training with only synthetic data.

In future works, the domain adaptation method can be further optimized by experimenting with choosing different shallow layers for extracting the local features for alignment. The optimization of other hyperparameters in the method like γ can also be performed. Future studies can also test and compare multiple SOTA domain adaptation techniques with various pose estimation models. It's also worth exploring other training strategies to incorporate synthetic data into the training of a network. For example, Riegler *et al.* [70] convert the data from both domains into one common domain and train the CNN on the larger synthetic dataset and fine-tune on the smaller real dataset. Furthermore, some different approaches to generating synthetic data can also be tried to enhance performance specifically on the SLP dataset. For example, Clever *et al.* [12] generate the poses in their synthetic depth data by fitting a 3D SMPL body model [53] to the poses in the SLP dataset. A different approach can also be taken by utilizing text-to-image diffusion models like ControlNet [95] to introduce synthetic occlusions to the existing human pose estimation datasets. Other such approaches can further improve the solution for the task and are left for future work.

The study highlights the effectiveness of synthetic data in solving a significant problem of pose estimation in horizontal positions under heavy occlusion faced by the current methods. While real datasets enhance the model's generalization ability to real-world applications, this research proves that the need for such extensive real datasets can be reduced to a good extent by utilizing synthetic data. Additionally, it also shows the potential of domain adaptation methods in utilizing

unlabeled real images to bridge the domain gap. In the broader sense of computer vision, this thesis demonstrates the potential of synthetic data in solving tasks that typically lack training data. This thesis also exemplifies how the real-world knowledge from pre-trained models can be utilized through retraining with synthetic data for specific tasks. It provides valuable insights into creating an effective synthetic dataset for training a CNN model.

Bibliography

[1] ABU ALHAIJA, Hassan ; MUSTIKOVELA, Siva K. ; MESCHEDER, Lars ; GEIGER, Andreas ; ROTHER, Carsten: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. In: *International Journal of Computer Vision* 126 (2018), S. 961–972

[2] ACHILLES, Felix ; ICHIM, Alexandru-Eugen ; COSKUN, Huseyin ; TOMBARI, Federico ; NOACHTAR, Soheyl ; NAVAB, Nassir: Patient MoCap: Human pose estimation under blanket occlusion for hospital monitoring applications. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part I 19* Springer, 2016, S. 491–499

[3] ANDRILUKA, Mykhaylo ; IQBAL, Umar ; INSAFUTDINOV, Eldar ; PISHCHULIN, Leonid ; MILAN, Anton ; GALL, Juergen ; SCHIELE, Bernt: Posetrack: A benchmark for human pose estimation and tracking. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, S. 5167–5176

[4] ANDRILUKA, Mykhaylo ; PISHCHULIN, Leonid ; GEHLER, Peter ; SCHIELE, Bernt: 2d human pose estimation: New benchmark and state of the art analysis. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, S. 3686–3693

[5] BOCHKOVSKIY, Alexey ; WANG, Chien-Yao ; LIAO, Hong-Yuan M.: Yolov4: Optimal speed and accuracy of object detection. In: *arXiv preprint arXiv:2004.10934* (2020)

[6] CAO, Zhe ; SIMON, Tomas ; WEI, Shih-En ; SHEIKH, Yaser: Realtime multi-person 2d pose estimation using part affinity fields. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 7291–7299

[7] CASAS, Leslie ; NAVAB, Nassir ; DEMIRCI, Stefanie: Patient 3D body pose estimation from pressure imaging. In: *International journal of computer assisted radiology and surgery* 14 (2019), S. 517–524

[8] CHEN, Haoming ; FENG, Runyang ; WU, Sifan ; XU, Hao ; ZHOU, Fengcheng ; LIU, Zhenguang: 2D Human pose estimation: A survey. In: *Multimedia Systems* 29 (2023), Nr. 5, S. 3115–3138

[9] CHEN, Yilun ; WANG, Zhicheng ; PENG, Yuxiang ; ZHANG, Zhiqiang ; YU, Gang ; SUN, Jian: Cascaded pyramid network for multi-person pose estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, S. 7103–7112

[10] CHENG, Bowen ; XIAO, Bin ; WANG, Jingdong ; SHI, Honghui ; HUANG, Thomas S. ; ZHANG, Lei: Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation.

In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, S. 5386–5395

[11] CLEVER, Henry: *BodyPressureSD*. http://dx.doi.org/10.7910/DVN/C6J1SP. Version: 2021

[12] CLEVER, Henry M. ; GRADY, Patrick L. ; TURK, Greg ; KEMP, Charles C.: BodyPressure - Inferring Body Pose and Contact Pressure From a Depth Image. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2023), Nr. 1, S. 137–153. http://dx.doi.org/10.1109/TPAMI.2022.3158902. – DOI 10.1109/TPAMI.2022.3158902

[13] CLEVER, Henry M. ; KAPUSTA, Ariel ; PARK, Daehyung ; ERICKSON, Zackory ; CHITALIA, Yash ; KEMP, Charles C.: 3d human pose estimation on a configurable bed from a pressure image. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2018, S. 54–61

[14] CORDTS, Marius ; OMRAN, Mohamed ; RAMOS, Sebastian ; REHFELD, Timo ; ENZWEILER, Markus ; BENENSON, Rodrigo ; FRANKE, Uwe ; ROTH, Stefan ; SCHIELE, Bernt: The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 3213–3223

[15] DALAL, Navneet ; TRIGGS, Bill: Histograms of oriented gradients for human detection. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* Bd. 1 Ieee, 2005, S. 886–893

[16] DANG, Qi ; YIN, Jianqin ; WANG, Bin ; ZHENG, Wenqing: Deep learning based 2d human pose estimation: A survey. In: *Tsinghua Science and Technology* 24 (2019), Nr. 6, S. 663–676

[17] DAVOODNIA, Vandad ; GHORBANI, Saeed ; ETEMAD, Ali: In-bed pressure-based pose estimation using image space representation learning. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE, 2021, S. 3965–3969

[18] DESMARAIS, Yann ; MOTTET, Denis ; SLANGEN, Pierre ; MONTESINOS, Philippe: A review of 3D human pose estimation algorithms for markerless motion capture. In: *Computer Vision and Image Understanding* 212 (2021), 103275. http://dx.doi.org/https://doi.org/10.1016/j.cviu.2021.103275. – DOI https://doi.org/10.1016/j.cviu.2021.103275. – ISSN 1077–3142

[19] DOSOVITSKIY, Alexey ; BEYER, Lucas ; KOLESNIKOV, Alexander ; WEISSENBORN, Dirk ; ZHAI, Xiaohua ; UNTERTHINER, Thomas ; DEHGHANI, Mostafa ; MINDERER, Matthias ; HEIGOLD, Georg ; GELLY, Sylvain u. a.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *arXiv preprint arXiv:2010.11929* (2020)

[20] EVERINGHAM, M. ; VAN GOOL, L. ; WILLIAMS, C. K. I. ; WINN, J. ; ZISSERMAN, A.: The Pascal Visual Object Classes (VOC) Challenge. In: *International Journal of Computer Vision* 88 (2010), Juni, Nr. 2, S. 303–338

[21] FABBRI, Matteo ; BRASÓ, Guillem ; MAUGERI, Gianluca ; CETINTAS, Orcun ; GASPARINI, Riccardo ; OŠEP, Aljoša ; CALDERARA, Simone ; LEAL-TAIXÉ, Laura ; CUCCHIARA, Rita: Motsynth: How can synthetic data help pedestrian detection and tracking? In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, S. 10849–10859

[22] FELZENSZWALB, Pedro ; MCALLESTER, David ; RAMANAN, Deva: A discriminatively trained, multiscale, deformable part model. In: *2008 IEEE conference on computer vision and pattern recognition* Ieee, 2008, S. 1–8

[23] FUKUSHIMA, Kunihiko: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. In: *Biological cybernetics* 36 (1980), Nr. 4, S. 193–202

[24] GAIDON, Adrien ; WANG, Qiao ; CABON, Yohann ; VIG, Eleonora: Virtual worlds as proxy for multi-object tracking analysis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 4340–4349

[25] GANIN, Yaroslav ; USTINOVA, Evgeniya ; AJAKAN, Hana ; GERMAIN, Pascal ; LAROCHELLE, Hugo ; LAVIOLETTE, François ; MARCHAND, Mario ; LEMPITSKY, Victor: Domain-adversarial training of neural networks. In: *The journal of machine learning research* 17 (2016), Nr. 1, S. 2096–2030

[26] GE, Zheng ; LIU, Songtao ; WANG, Feng ; LI, Zeming ; SUN, Jian: Yolox: Exceeding yolo series in 2021. In: *arXiv preprint arXiv:2107.08430* (2021)

[27] GEIGER, Andreas ; LENZ, Philip ; STILLER, Christoph ; URTASUN, Raquel: Vision meets robotics: The kitti dataset. In: *The International Journal of Robotics Research* 32 (2013), Nr. 11, S. 1231–1237

[28] GIRSHICK, Ross: Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, 2015, S. 1440–1448

[29] GIRSHICK, Ross ; DONAHUE, Jeff ; DARRELL, Trevor ; MALIK, Jitendra: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, S. 580–587

[30] GIRSHICK, Ross ; DONAHUE, Jeff ; DARRELL, Trevor ; MALIK, Jitendra: Region-based convolutional networks for accurate object detection and segmentation. In: *IEEE transactions on pattern analysis and machine intelligence* 38 (2015), Nr. 1, S. 142–158

[31] GOODFELLOW, Ian ; POUGET-ABADIE, Jean ; MIRZA, Mehdi ; XU, Bing ; WARDE-FARLEY, David ; OZAIR, Sherjil ; COURVILLE, Aaron ; BENGIO, Yoshua: Generative adversarial networks. In: *Communications of the ACM* 63 (2020), Nr. 11, S. 139–144

[32] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *IEEE transactions on pattern analysis and machine intelligence* 37 (2015), Nr. 9, S. 1904–1916

[33] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 770–778

[34] HUANG, Rui ; ZHANG, Shu ; LI, Tianyu ; HE, Ran: Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In: *Proceedings of the IEEE international conference on computer vision*, 2017, S. 2439–2448

[35] IONESCU, Catalin ; PAPAVA, Dragos ; OLARU, Vlad ; SMINCHISESCU, Cristian: Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (2014), jul, Nr. 7, S. 1325–1339

[36] JOHNSON, Sam ; EVERINGHAM, Mark: Clustered pose and nonlinear appearance models for human pose estimation. In: *bmvc* Bd. 2 Aberystwyth, UK, 2010, S. 5

[37] KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; HINTON, Geoffrey E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems* 25 (2012)

[38] KYROLLOS, Daniel G. ; FULLER, Anthony ; GREENWOOD, Kim ; HARROLD, JoAnn ; GREEN, James R.: Under the cover infant pose estimation using multimodal data. In: *IEEE Transactions on Instrumentation and Measurement* 72 (2023), S. 1–12

[39] KYROLLOS, Daniel G. ; HASSAN, Randa ; DOSSO, Yasmina S. ; GREEN, James R.: Fusing pressure-sensitive mat data with video through multi-modal registration. In: *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* IEEE, 2021, S. 1–6

[40] LAN, Gongjin ; WU, Yu ; HU, Fei ; HAO, Qi: Vision-based human pose estimation via deep learning: A survey. In: *IEEE Transactions on Human-Machine Systems* (2022)

[41] LECUN, Yann ; BOTTOU, Léon ; BENGIO, Yoshua ; HAFFNER, Patrick: Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324

[42] LI, Jiefeng ; WANG, Can ; ZHU, Hao ; MAO, Yihuan ; FANG, Hao-Shu ; LU, Cewu: Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, S. 10863–10872

[43] LI, Ke ; WANG, Shijie ; ZHANG, Xiang ; XU, Yifan ; XU, Weijian ; TU, Zhuowen: Pose recognition with cascade transformers. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, S. 1944–1953

[44] LI, Xiang ; ZHANG, Wei ; DING, Qian ; SUN, Jian-Qiao: Multi-layer domain adaptation method for rolling bearing fault diagnosis. In: *Signal processing* 157 (2019), S. 180–197

[45] LIN, Tsung-Yi ; GOYAL, Priya ; GIRSHICK, Ross ; HE, Kaiming ; DOLLÁR, Piotr: Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*, 2017, S. 2980–2988

[46] LIN, Tsung-Yi ; MAIRE, Michael ; BELONGIE, Serge ; HAYS, James ; PERONA, Pietro ; RAMANAN, Deva ; DOLLÁR, Piotr ; ZITNICK, C L.: Microsoft coco: Common objects in context. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* Springer, 2014, S. 740–755

[47] LIU, Shu ; QI, Lu ; QIN, Haifang ; SHI, Jianping ; JIA, Jiaya: Path aggregation network for instance segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, S. 8759–8768

[48] LIU, Shuangjun ; HUANG, Xiaofei ; FU, Nihang ; LI, Cheng ; SU, Zhongnan ; OSTADABBAS, Sarah: Simultaneously-collected multimodal lying pose dataset: Enabling in-bed human pose monitoring. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2022), Nr. 1, S. 1106–1118

[49] LIU, Shuangjun ; OSTADABBAS, Sarah: A vision-based system for in-bed posture tracking. In: *Proceedings of the IEEE international conference on computer vision workshops*, 2017, S. 1373–1382

[50] LIU, Shuangjun ; OSTADABBAS, Sarah: Seeing under the cover: A physics guided learning approach for in-bed pose estimation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention* Springer, 2019, S. 236–245

[51] LIU, Shuangjun ; YIN, Yu ; OSTADABBAS, Sarah: In-Bed Pose Estimation: Deep Learning With Shallow Dataset. In: *IEEE Journal of Translational Engineering in Health and Medicine* 7 (2019), S. 1–12. http://dx.doi.org/10.1109/JTEHM.2019.2892970. – DOI 10.1109/JTEHM.2019.2892970

[52] LIU, Wei ; ANGUELOV, Dragomir ; ERHAN, Dumitru ; SZEGEDY, Christian ; REED, Scott ; FU, Cheng-Yang ; BERG, Alexander C.: Ssd: Single shot multibox detector. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* Springer, 2016, S. 21–37

[53] LOPER, Matthew ; MAHMOOD, Naureen ; ROMERO, Javier ; PONS-MOLL, Gerard ; BLACK, Michael J.: SMPL: A Skinned Multi-Person Linear Model. In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34 (2015), Oktober, Nr. 6, S. 248:1–248:16

[54] LOVANSHI, Mayank ; TIWARI, Vivek: Human pose estimation: benchmarking deep learning-based methods. In: *2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)* IEEE, 2022, S. 1–6

[55] MAJI, Debapriya ; NAGORI, Soyeb ; MATHEW, Manu ; PODDAR, Deepak: Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, S. 2637–2646

[56] MAN, Keith ; CHAHL, Javaan: A Review of Synthetic Image Data and Its Use in Computer Vision. In: *Journal of Imaging* 8 (2022), Nr. 11, S. 310

[57] MAO, Weian ; GE, Yongtao ; SHEN, Chunhua ; TIAN, Zhi ; WANG, Xinlong ; WANG, Zhibin: Tfpose: Direct human pose estimation with transformers. In: *arXiv preprint arXiv:2103.15320* (2021)

[58] MOHAMMADI, Sara M. ; ENSHAEIFAR, Shirin ; HILTON, Adrian ; DIJK, Derk-Jan ; WELLS, Kevin: Transfer Learning for Clinical Sleep Pose Detection Using a Single 2D IR Camera. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering*

29 (2021), S. 290–299. http://dx.doi.org/10.1109/TNSRE.2020.3048121. – DOI 10.1109/TNSRE.2020.3048121

[59] MUNEA, Tewodros L. ; JEMBRE, Yalew Z. ; WELDEGEBRIEL, Halefom T. ; CHEN, Longbiao ; HUANG, Chenxi ; YANG, Chenhui: The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation. In: *IEEE Access* 8 (2020), S. 133330–133348

[60] NEFF, Christopher ; SHETH, Aneri ; FURGURSON, Steven ; TABKHI, Hamed: Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation. In: *arXiv preprint arXiv:2007.08090* (2020)

[61] NGUYEN, Thong D. ; KRESOVIC, Milan: A survey of top-down approaches for human pose estimation. In: *arXiv preprint arXiv:2202.02656* (2022)

[62] NIKOLENKO, Sergey I.: *Synthetic data for deep learning*. Bd. 174. Springer, 2021

[63] NOWRUZI, Farzan E. ; KAPOOR, Prince ; KOLHATKAR, Dhanvin ; HASSANAT, Fahed A. ; LAGANIERE, Robert ; REBUT, Julien: How much real data do we actually need: Analyzing object detection performance using synthetic and real data. In: *arXiv preprint arXiv:1907.07061* (2019)

[64] OLAH, Chris ; MORDVINTSEV, Alexander ; SCHUBERT, Ludwig: Feature visualization. In: *Distill* 2 (2017), Nr. 11, S. e7

[65] PURKRÁBEK, Miroslav ; MATAS, Jiří: Improving 2D Human Pose Estimation across Unseen Camera Views with Synthetic Data. In: *arXiv preprint arXiv:2307.06737* (2023)

[66] REDMON, Joseph ; DIVVALA, Santosh ; GIRSHICK, Ross ; FARHADI, Ali: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 779–788

[67] REDMON, Joseph ; FARHADI, Ali: YOLO9000: better, faster, stronger. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 7263–7271

[68] REDMON, Joseph ; FARHADI, Ali: Yolov3: An incremental improvement. In: *arXiv preprint arXiv:1804.02767* (2018)

[69] REN, Shaoqing ; HE, Kaiming ; GIRSHICK, Ross ; SUN, Jian: Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems* 28 (2015)

[70] RIEGLER, Gernot ; URSCHLER, Martin ; RUTHER, Matthias ; BISCHOF, Horst ; STERN, Darko: Anatomical landmark detection in medical applications driven by synthetic data. In: *Proceedings of the IEEE international conference on computer vision workshops*, 2015, S. 12–16

[71] ROBERTS, Lawrence: *MACHINE PERCEPTION OF THREE-DIMENSIONAL, SOLIDS*, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, Diss., 1963

[72] ROS, German ; SELLART, Laura ; MATERZYNSKA, Joanna ; VAZQUEZ, David ; LOPEZ, Antonio M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 3234–3243

[73] RUGGERO RONCHI, Matteo ; PERONA, Pietro: Benchmarking and error diagnosis in multi-instance pose estimation. In: *Proceedings of the IEEE international conference on computer vision*, 2017, S. 369–378

[74] RUSSAKOVSKY, Olga ; DENG, Jia ; SU, Hao ; KRAUSE, Jonathan ; SATHEESH, Sanjeev ; MA, Sean ; HUANG, Zhiheng ; KARPATHY, Andrej ; KHOSLA, Aditya ; BERNSTEIN, Michael u. a.: Imagenet large scale visual recognition challenge. In: *International journal of computer vision* 115 (2015), S. 211–252

[75] SAITO, Kuniaki ; USHIKU, Yoshitaka ; HARADA, Tatsuya ; SAENKO, Kate: Strong-weak distribution alignment for adaptive object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, S. 6956–6965

[76] SAKARIDIS, Christos ; DAI, Dengxin ; VAN GOOL, Luc: Semantic foggy scene understanding with synthetic data. In: *International Journal of Computer Vision* 126 (2018), S. 973–992

[77] SARAFIANOS, Nikolaos ; BOTEANU, Bogdan ; IONESCU, Bogdan ; KAKADIARIS, Ioannis A.: 3d human pose estimation: A review of the literature and analysis of covariates. In: *Computer Vision and Image Understanding* 152 (2016), S. 1–20

[78] SHACHAF, Gal ; BRUTZKUS, Alon ; GLOBERSON, Amir: A theoretical analysis of fine-tuning with linear teachers. In: *Advances in Neural Information Processing Systems* 34 (2021), S. 15382–15394

[79] SHRIVASTAVA, Ashish ; PFISTER, Tomas ; TUZEL, Oncel ; SUSSKIND, Joshua ; WANG, Wenda ; WEBB, Russell: Learning from simulated and unsupervised images through adversarial training. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 2107–2116

[80] SIMONYAN, Karen ; ZISSERMAN, Andrew: Very deep convolutional networks for large-scale image recognition. In: *arXiv preprint arXiv:1409.1556* (2014)

[81] SZEGEDY, Christian ; LIU, Wei ; JIA, Yangqing ; SERMANET, Pierre ; REED, Scott ; ANGUELOV, Dragomir ; ERHAN, Dumitru ; VANHOUCKE, Vincent ; RABINOVICH, Andrew: Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, S. 1–9

[82] TOBIN, Josh ; FONG, Rachel ; RAY, Alex ; SCHNEIDER, Jonas ; ZAREMBA, Wojciech ; ABBEEL, Pieter: Domain randomization for transferring deep neural networks from simulation to the real world. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* IEEE, 2017, S. 23–30

[83] TOMMASI, Tatiana ; ORABONA, Francesco ; CAPUTO, Barbara: Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* IEEE, 2010, S. 3081–3088

[84] TREMBLAY, Jonathan ; PRAKASH, Aayush ; ACUNA, David ; BROPHY, Mark ; JAMPANI, Varun ; ANIL, Cem ; TO, Thang ; CAMERACCI, Eric ; BOOCHOON, Shaad ; BIRCHFIELD, Stan: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, S. 969–977

[85] VAROL, Gul ; ROMERO, Javier ; MARTIN, Xavier ; MAHMOOD, Naureen ; BLACK, Michael J. ; LAPTEV, Ivan ; SCHMID, Cordelia: Learning from synthetic humans. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 109–117

[86] VIOLA, Paul ; JONES, Michael: Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* Bd. 1 Ieee, 2001, S. I–I

[87] VIOLA, Paul ; JONES, Michael J.: Robust real-time face detection. In: *International journal of computer vision* 57 (2004), S. 137–154

[88] VOULODIMOS, Athanasios ; DOULAMIS, Nikolaos ; DOULAMIS, Anastasios ; PROTOPAPADAKIS, Eftychios u. a.: Deep learning for computer vision: A brief review. In: *Computational intelligence and neuroscience* 2018 (2018)

[89] WANG, Chien-Yao ; BOCHKOVSKIY, Alexey ; LIAO, Hong-Yuan M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *arXiv preprint arXiv:2207.02696* (2022)

[90] WANG, Chien-Yao ; LIAO, Hong-Yuan M. ; WU, Yueh-Hua ; CHEN, Ping-Yang ; HSIEH, Jun-Wei ; YEH, I-Hau: CSPNet: A new backbone that can enhance learning capability of CNN. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, S. 390–391

[91] WOOD, Erroll ; BALTRUŠAITIS, Tadas ; HEWITT, Charlie ; DZIADZIO, Sebastian ; CASHMAN, Thomas J. ; SHOTTON, Jamie: Fake it till you make it: face analysis in the wild using synthetic data alone. In: *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, S. 3681–3691

[92] XU, Yufei ; ZHANG, Jing ; ZHANG, Qiming ; TAO, Dacheng: Vitpose: Simple vision transformer baselines for human pose estimation. In: *Advances in Neural Information Processing Systems* 35 (2022), S. 38571–38584

[93] YANG, Sen ; QUAN, Zhibin ; NIE, Mu ; YANG, Wankou: Transpose: Keypoint localization via transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, S. 11802–11812

[94] YU, Zefang ; LI, Yangcheng ; LIU, Yicheng ; LIU, Ting ; FU, Yuzhuo: Synpose: A large-scale and densely annotated synthetic dataset for human pose estimation in classroom. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE, 2022, S. 3428–3432

[95] ZHANG, Lvmin ; RAO, Anyi ; AGRAWALA, Maneesh: Adding conditional control to text-to-image diffusion models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, S. 3836–3847

[96] ZHANG, Song-Hai ; LI, Ruilong ; DONG, Xin ; ROSIN, Paul ; CAI, Zixi ; HAN, Xi ; YANG, Dingcheng ; HUANG, Haozhi ; HU, Shi-Min: Pose2seg: Detection free human instance segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, S. 889–898

[97] ZHANG, Weiyu ; ZHU, Menglong ; DERPANIS, Konstantinos G.: From actemes to action: A strongly-supervised representation for detailed action understanding. In: *Proceedings of the IEEE international conference on computer vision*, 2013, S. 2248–2255

[98] ZOU, Zhengxia ; CHEN, Keyan ; SHI, Zhenwei ; GUO, Yuhong ; YE, Jieping: Object detection in 20 years: A survey. In: *Proceedings of the IEEE* (2023)

Glossary

Epoch A complete pass of the training dataset through the machine learning algorithm. 36, 44, 47, 48

HSV Hue, Saturation and Value. An alternative representation of the RGB color model. 29, 30

Learning Rate It represents the step size at which a model's parameters are updated during training. 44, 48, 51, 56

MoCap Motion capture: The process of recording the movement of objects or people and translating that data into a digital form. 22

PyTorch A machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing. 36

RGB An image with red, green and blue channels. vii, 1, 2, 4, 11, 20, 27, 38, 61

Sim2Real It refers to techniques that can be used to transfer the knowledge acquired from a simulated environment to a real environment. 23, 37, 39, 56, 62

SOTA State-of-the-Art: It refers to the highest level of general development achieved at a particular time. vii, 4, 7, 11–13, 15–17, 22, 35, 37, 43, 57, 62

Warmup Epoch An initial training phase with a lower learning rate, gradually increasing to the planned rate, designed to stabilize the model at the beginning of the training process. 44

Acronyms

ANN Artificial Neural Network 7, 8, 15, 38

API Application Programming Interface 3

CNN Convolutional Neural Network v, 3, 7–10, 15, 16, 21–23, 36, 40, 61–63

COCO Common Objects in Context vii, ix, 13, 14, 16, 18–20, 23, 25, 27, 36, 37, 62, 84

CV Computer Vision v, 1–3, 5, 7–9, 11, 21, 22, 36, 56

DA Domain Adaptation vi, 5, 23, 35, 39, 41, 43, 56

DPM Deformable Part-Based Model 10

DR Domain Randomization v, 23, 28, 30, 34

GAN Generative Adversarial Networks 23

GPU Graphics Processing Unit 7

HDRI High Dynamic Range Imaging 25, 29, 30

HOG Histogram of Oriented Gradients 10

HPE Human Pose Estimation v, ix, 1, 4, 5, 7, 8, 10–20, 22, 24, 27, 28, 30, 31, 35–37, 43, 47, 49, 57

IoU Intersection over Union vii, 13, 14, 17

IR Infrared 4, 20

mAP Mean Average Precision 13, 15, 19, 36, 37, 43–45, 47–54, 56–58, 85, 88

MPJPE Mean Per Joint Position Error 13

MPJVE Mean Per Joint Velocity Error 13

OKS Object Keypoint Similarity vi, vii, ix, 13–15, 17, 37, 38, 52

PAF Part Affinity Fields 17

PCK Percentage of Correct Keypoints 13

PCP Percentage of Correct Parts 13

RCNN Regions with CNN features 10, 11, 39

RPN Region Proposal Network 10, 39, 40

SLP Simultaneously-Collected Multimodal Lying Pose vi–ix, 20, 21, 26–28, 30, 31, 35, 38, 41, 43–48, 51–54, 56–59, 61, 62, 79–83, 86, 87

SMPL Skinned Multi-Person Linear Model 62

SSD Single-Shot Detector 10, 11, 36

SVM Support Vector Machines 7, 10

SWDA Strong-Weak Distribution Alignment vi, vii, 39–41, 56, 57

TL Transfer Learning vi, 4, 36, 37, 47, 49

YOLO You Only Look Once vii, 10, 11, 18, 19, 25, 26, 36, 40, 61

Appendix

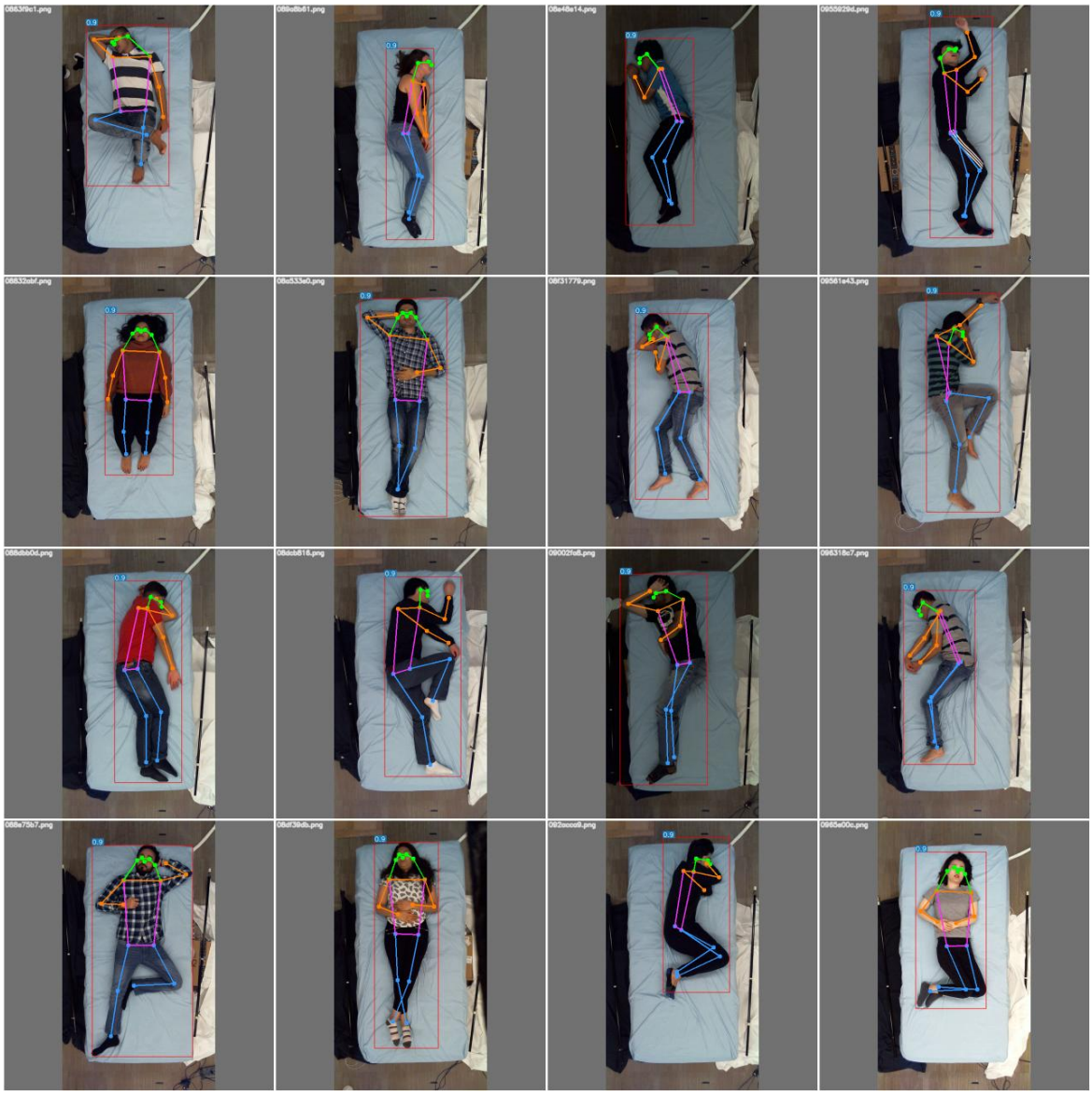


Figure 1: Inference examples of pre-trained YOLOv7 on the (*uncover*) set of SLP

Figure 2: Inference examples of pre-trained YOLOv7 on SLP (*cover1*) showing that only the uncovered keypoints and bounding boxes are predicted by the model.

Figure 3: Inference examples of pre-trained YOLOv7 on SLP (*cover2*) showing that only the uncovered keypoints and bounding boxes are predicted by the model.

Figure 4: Examples from the *cover1* and *cover2* set of SLP dataset [48] showing the severe lack of visual information about the joints of the human body under the blanket. The models retrained in the thesis with synthetic data can still predict the poses under the blanket up to a good extent. This also demonstrates the difficulty of manually annotating such datasets without sophisticated equipment.

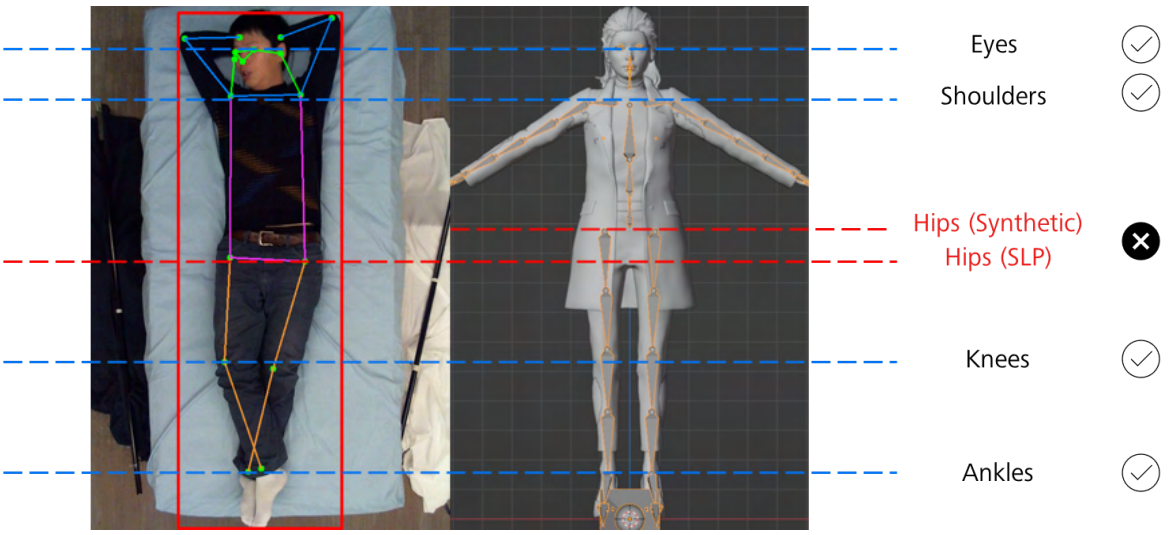


Figure 5: Difference in annotations of the synthetic datasets as compared to the SLP dataset [48]. The hip joint in the SLP dataset is considered to be in an unusually low position on the body. This difference is ignored in the thesis. Some metrics and predictions suffer because of this difference.

Figure 6: Problems with blanket simulation in the RandPose dataset

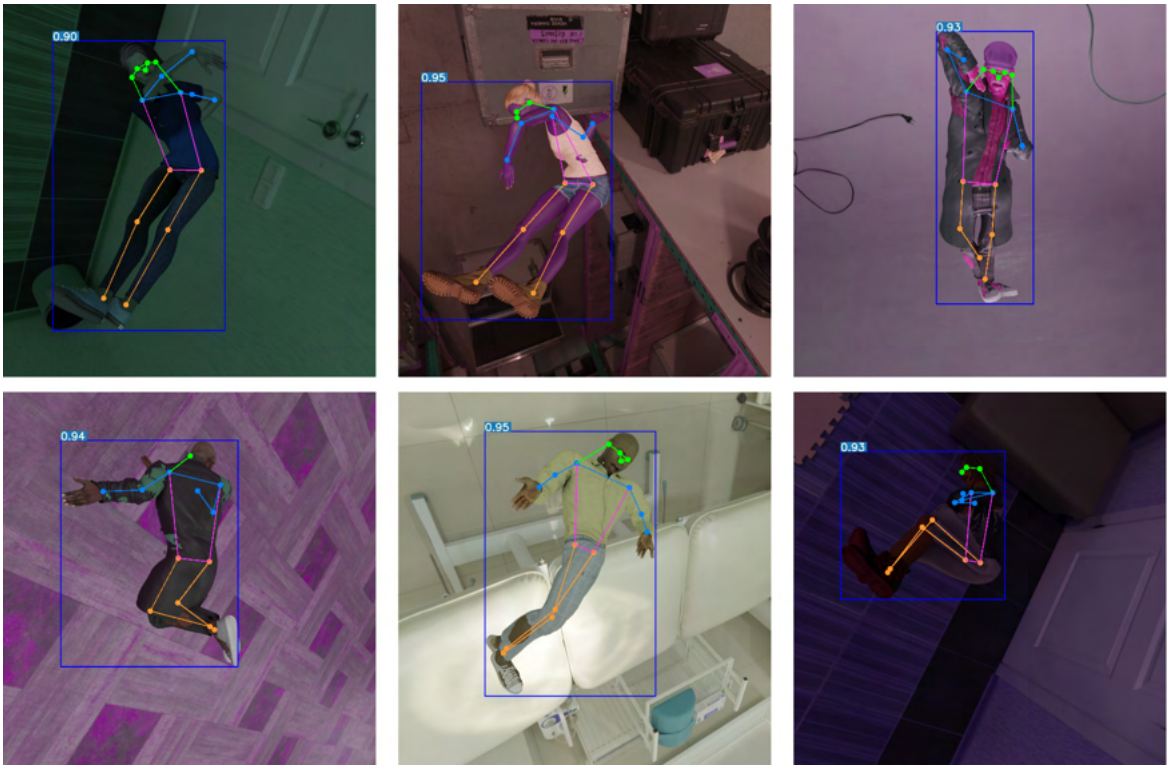


Figure 7: Inference of pre-trained YOLOv7 on the generated synthetic dataset. The results show a decent performance of the models pre-trained with the COCO dataset on the uncovered synthetic humans.

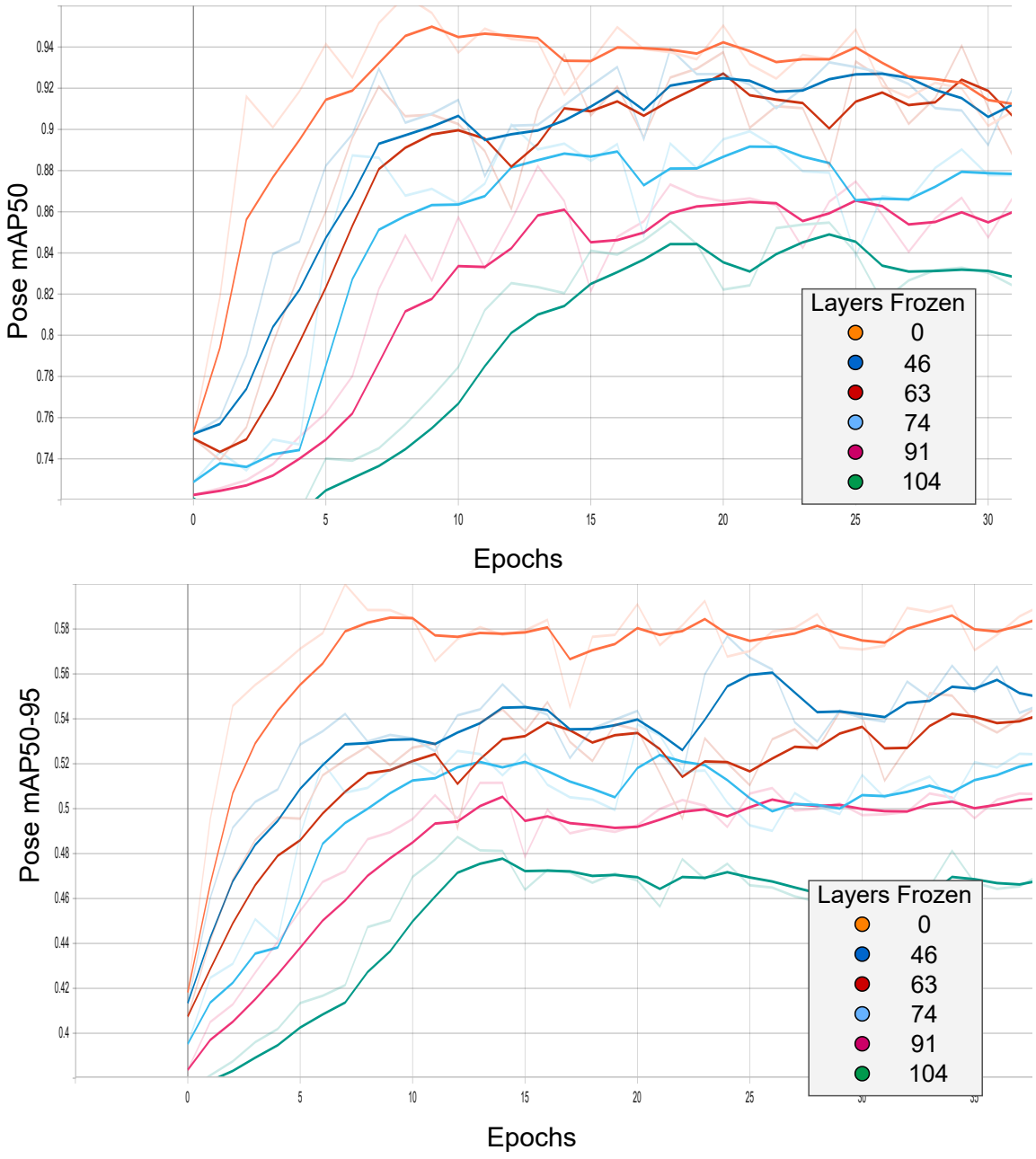


Figure 8: Pose mAP_{50} (top) and mAP_{50-95} (bottom) graphs showing accuracy with freezing different number of layers while training YOLOv7 on 500 images.

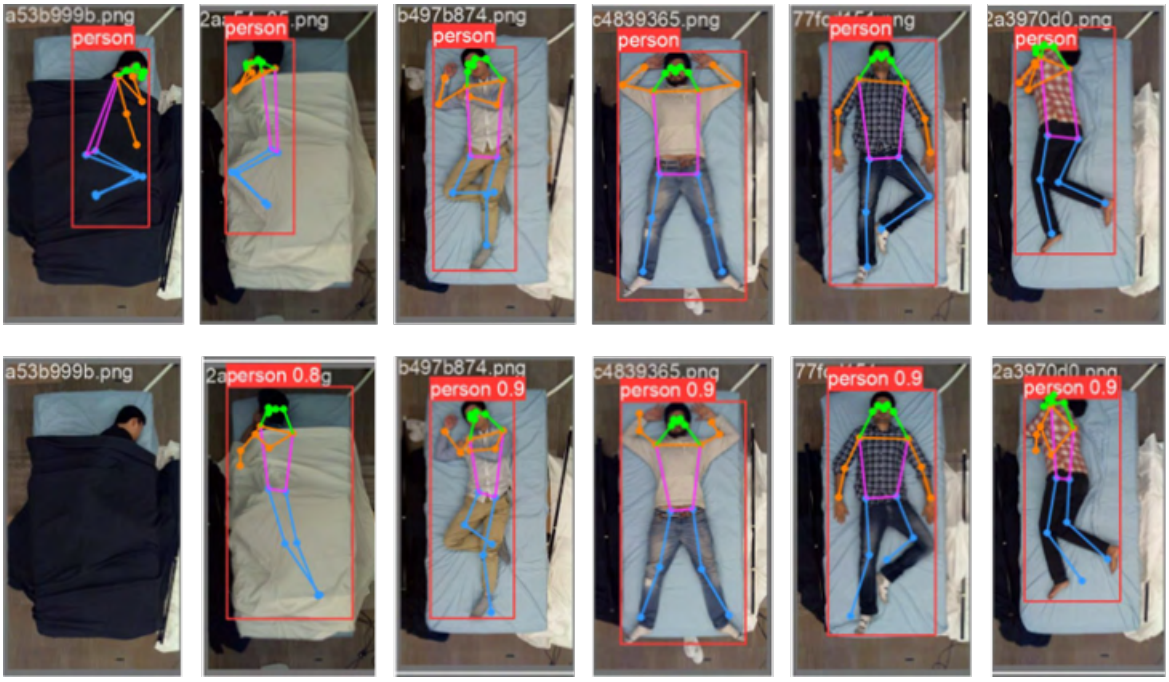


Figure 9: Inference examples of re-trained YOLOv8 on SLP (*cover* set) with ground truth (top) and predictions (bottom). YOLOv8 is re-trained with 500 images from the RealPose dataset without freezing any layers for 30 epochs. The results show missing bounding box predictions and wrong keypoint predictions even on the uncovered joints. This demonstrates the poor training process of YOLOv8 with the synthetic datasets.

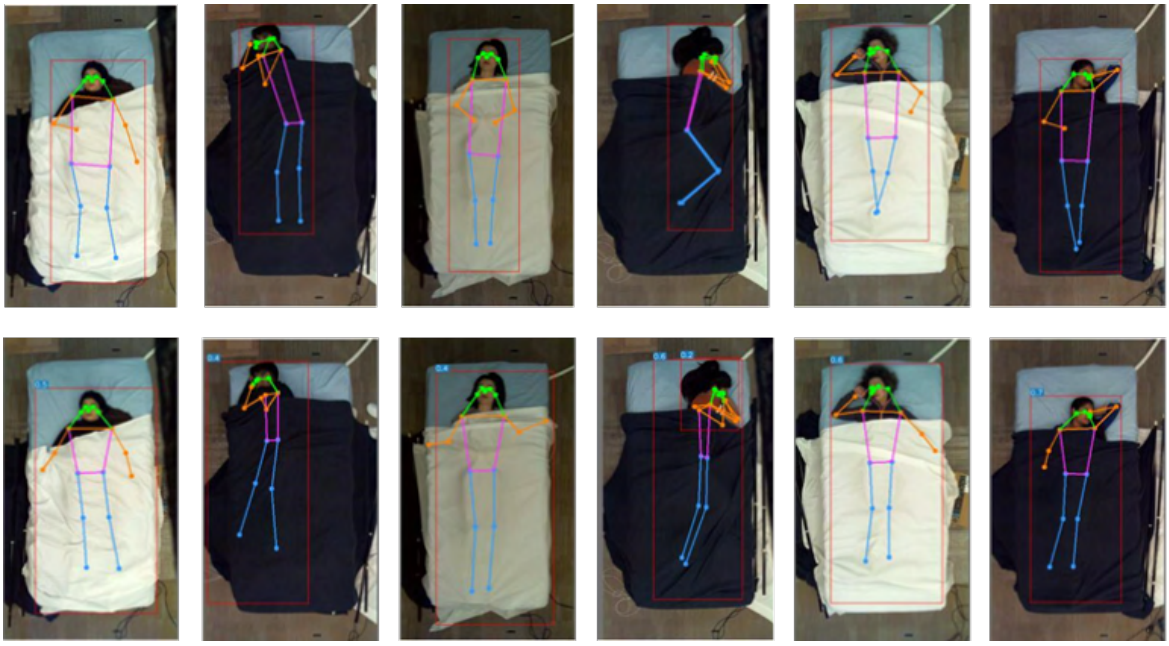


Figure 10: Inference examples of re-trained YOLOv7 on SLP (*cover* set) with ground truth (top) and predictions (bottom). YOLOv7 is re-trained with 500 images from the RealPose dataset without freezing any layers for 30 epochs. The results show significant improvements over the pre-trained model with correct bounding box predictions and decent keypoint predictions. Some extreme poses could still not be predicted with precision after training with only synthetic data. Most of the images show a constant error in the prediction of the hip joints. The cause of this error is due to the difference in the annotation location of hips as shown in Fig. 5.

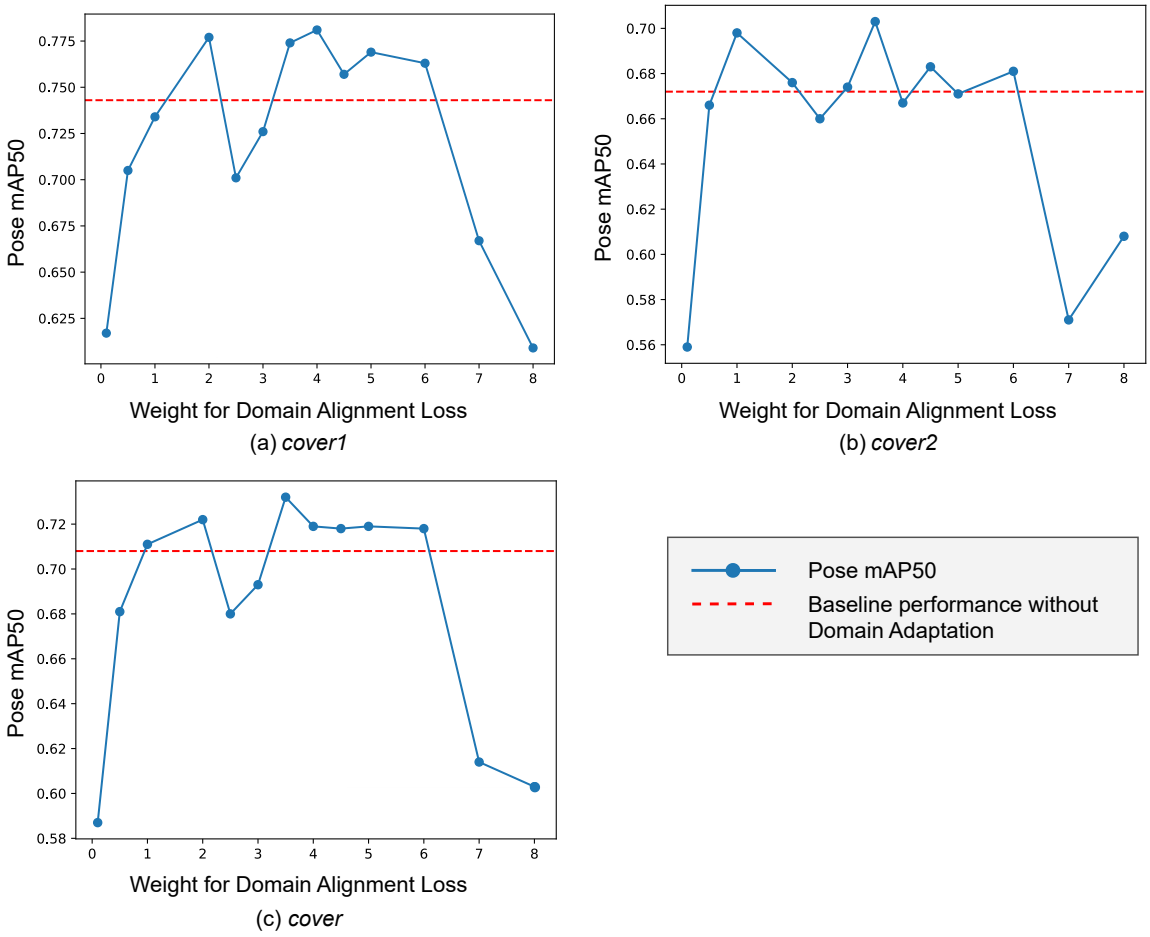


Figure 11: Results of varying the weight for domain alignment loss in YOLOv7-DA on the pose mAP_{50} with **RealPose** shown for *cover1* in (a), *cover2* in (b) and *cover* in (c). The baseline performance is the best performance of YOLOv7 retrained with the RealPose dataset without domain adaptation shown in Table 5.3.