Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Comparing Synthetic and Real Data for Anthropometric Measurements Estimation

Bachelor Thesis

2024
Michal Baránek

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Comparing Synthetic and Real Data for Anthropometric Measurements Estimation

Bachelor Thesis

Study Programme:	Computer Science
Field of Study:	Computer Science
Department:	Department of Computer Science
Supervisor:	Mgr. Dana Škorvánková

Bratislava, 2024
Michal Baránek

18794588

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Michal Baránek
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Comparing Synthetic and Real Data for Anthropometric Measurements Estimation
Porovnanie syntetických a reálnych dát pre účely odhadu antropometrických mier

Anotácia: Odhad mier ľudského tela je úloha, ktorá priťahuje v posledných rokoch pozornosť viacerých vedeckých oblastí. Automatický a presný prístup na riešenie tohto problému je kľúčový v rôznych oblastiach počítačového videnia. Výroba odevov a šitie odevov na mieru patria medzi aplikácie, kde by presný odhad mier tela z vizuálnych dát človeka bol prínosným, nahradením tradičného manuálneho merania tela.

Cieľ: Cieľom tejto bakalárskej práce je naštudovať problematiku odhadu rozmerov ľudského tela, otestovať vybrané state-of-the-art metódy založené na hlbokom učení a evaluovať ich pomocou syntetických aj reálnych dát. Úlohou je týmto spôsobom analyzovať rozdiely medzi danými doménami, a vyhodnotiť výhody augmentácie reálnych dát pomocou syntetických obrazov pre účely trénovania modelu.

Kľúčové slová: miery tela, neurónové siete, hlboké učenie

Vedúci: Mgr. Dana Škorvánková
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Dátum zadania: 25.09.2023

Dátum schválenia: 04.10.2023

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

....
študent

....
vedúci práce

18794588

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Michal Baránek
Study programme:	Applied Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study:	Computer Science
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Comparing Synthetic and Real Data for Anthropometric Measurements Estimation

Annotation: Human body measurements estimation is a task that attracts the attention of several scientific fields in recent years. An automatic and accurate approach to this problem is crucial in various areas of the computer vision-oriented industry. Garment manufacturing and tailoring are some of the applications, where an accurate body measurements estimation from visual human data would be beneficial, replacing the traditional manual tape measuring.

Aim: The goal of the bachelor thesis is to study the task of body measurements estimation. The aim is to test selected state-of-the-art deep learning methods and evaluate them using both synthetic and real human body data. In this way, we aim to analyze the domain gap and explore the benefits of augmenting real data with synthetic images for training purposes.

Keywords: body measurements, neural networks, deep learning

Supervisor:	Mgr. Dana Škorvánková
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	doc. RNDr. Tatiana Jajcayová, PhD.
Assigned:	25.09.2023
Approved:	04.10.2023

doc. RNDr. Damas Gruska, PhD.
Guarantor of Study Programme

....
Student

....
Supervisor

Acknowledgments: Tu môžete poďakovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Contents

List of Figures

List of Tables

Introduction

Chapter 1

Overview

The content of this chapter provides definitions and explanations of terms and methods used in the following chapters with intention to aid in understanding the topic discussed in this thesis.

1.1 Anthropometry

Our goal in this thesis is to estimate human body measurements. These are important for certain tasks as clothing sewing, virtual environment calibration, creating realistic avatars and more. For these task we need a set of measurements which will guide us. Measurement locations vary by the use, and thus there is no universal guide. Professionals should be familiar with the measurements required in their field of application, but the subjects themselves are usually not as informed. This can then result in incorrect measurement. Using a trained neural network model can prevent that. Another advantage of using a neural network is the speed of measurements. Compared to manual measurement, our neural network based approach can estimate multiple measurements at once, whereas while using the manual approach we only get one measurement at a time. This makes our approach easier to use than

The human body can be measured using different methods. These are usually dependent on input data, and thus not every method can be used in every situation.

1.1.1 Manual measurement

This is the traditional method of using tape or any similar measuring device to obtain measurements. The approach usually requires one extra person who performs the measurement on the subject. Due to measurements that must be taken at specific locations to provide correct information, a person without help is more prone to obtaining incorrect measurements.

1.1.2 Virtual measuring on 3D model

Another method used to obtain measurements is the use of scanning technology. One of the approaches uses devices such as photogrammetry scanners to create realistic meshes of the scanned subject. The data can then be registered to the mesh topology of SMPL [1]. We can then use the resulting mesh to calculate the required measurements. This is also the case for [2], which uses these measurements as ground-truth data. A different approach uses 3D scanning devices. These are more expensive than the equipment required for photogrammetry, but they can provide precise models. The result of a scan using a 3D scanner is a point cloud or a set of data points that represent the shape and size of the subject [3]. This can be used directly as input for further processing.

Photogrammetry scanner

Photogrammetry scanners utilise photographs from different angles and positions to calculate and create a mesh. The device to take photos does not need any extra functionality to produce images that are processable in the final model. However, photogrammetry scanners are being developed to provide ease of use and quality over common cameras. Scanners can be joined into a multi-camera system. This allows capturing photos from multiple angles simultaneously, which provides more accurate data compared to using a single camera setup.

3D scanner

Another type of scanner is the 3D scanner, which employs various techniques to measure the distance from the camera to the subject. Commonly used methods include structured light and laser-based scanning. Both techniques use light to determine the location of points on the subject. Laser-based scanners project lines onto the subject and record the reflections, while structured light scanners project a grid pattern. By analyzing the distortions in the reflected light, these devices can accurately measure distances and create detailed 3D models.

1.2 SMPL

SMPL, an acronym for Skinned Multi-Person Linear model [1], is a versatile tool for generating animated human bodies with diverse body shapes and specific poses. One of its notable features is its ability to simulate natural soft-tissue deformation, resulting in lifelike movements. This is achieved through the integration of blend shapes and joints.

Blend shapes in SMPL are defined as a vector of concatenated vertex offsets. Each blend shape is a predefined deformation of the mesh, which, when combined, can represent complex changes in body shape and pose. Essentially, these are sets of vertices that, when adjusted, morph the mesh to match various body forms and postures. By linearly blending these shapes, SMPL can produce a wide range of realistic human body shapes. By using blend shapes, the model can adapt to subtle changes in body shape caused by muscle movement, fat distribution, and individual anatomy.

The SMPL model incorporates a skeletal structure composed of joints that create a hierarchy creating a skeleton. This skeleton defines the kinematic chain used for pose transformations. Each joint is associated with a specific part of the body, and the movement of these joints dictates the overall pose of the model. The character's skeletal structure is rigged to allow for flexible posing while following anatomical correctness.

1.3 Neural Networks

A neural network is a computational model inspired by the structure and function of the human brain. It consists of interconnected nodes, called neurons, that work together to process information. Each neuron receives input variables, processes them, and passes the results to other neurons. These connections have associated weights that determine the influence of each input on the neuron's output.

Neurons are organized into layers: input layers, hidden layers, and output layers. The input layer receives the initial data, the hidden layers perform intermediate computations, and the output layer produces the final result. The hidden layers are crucial for transforming the input data into the desired output by performing complex operations.

Training a neural network is based on adjusting the weights of the connections. This process, called learning, typically involves comparing the network's output to the correct values (ground truth) and trying to minimize the difference, often through a method known as backpropagation. By iteratively adjusting the weights, the network learns to recognize patterns and relationships within the data that might be difficult for humans to find.

This ability to learn and generalize from data enables neural networks to solve a wide range of problems, from image recognition to natural language processing.

1.3.1 Hyperparameter

A hyperparameter is a parameter that is not learned but chosen by the developer. These parameters do not change over time. These can be - the choice of optimizer, learning rate, number of layers, filter size and more.

1.3.2 Loss functions

The training process is guided by the loss function [4]. They show how good or how bad the network is at predicting the output. The results then serve as a guide for the learnable parameters. The loss function measures the difference between the predicted and expected outputs. The main goal of the network then becomes to minimise the loss function. One of the commonly used losses for regression task such as this one is the mean squared error (MSE sometimes called L2 loss). This loss function is calculated as an average of the squared difference between the predicted values and the ground truth. To define this function mathematically:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.1}$$

In this equation n is used to define number of samples, y_i is the ground truth and $\hat{y}_i$ is the predicted value of the i^{th} sample. Being a convex function, the MSE has a unique global minimum, which helps the optimization process as the optimization methods do not get stuck in the local minima. While being computationally simple, it is vulnerable to outliers. The issue is created by the square nature of this function. In the case of an existing outlier, the function gets heavily influenced and may not perform as well.

1.3.3 Performance metrics

After the network is trained on the training data, it has to face new, unseen data. This ability is then measured using performance metrics. They are mainly used after the network has been trained. Performance metrics are also used to compare different networks. We can use some loss functions for performance metrics. In this thesis, we will use mean absolute error (MAE or L1 loss). The principle is similar to MSE, but instead of squaring the difference, we will use the absolute value to always have an error larger than 0, meaning 0 will be a perfect fit. The mathematical definition of the MAE is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{1.2}$$

in which n is used to define number of samples, y_i is the ground truth and $\hat{y}_i$ the predicted value of the i^{th} sample.

1.3.4 Overfitting

A significant challenge when using neural networks is the risk of overfitting. Overfitting occurs when a complex network is trained on an insufficient amount of data.

While the sophisticated architecture of a neural network enables it to capture intricate relationships within the data, a lack of sufficient training data can lead the network to memorize the exact values of the training examples rather than learning the underlying patterns. As a result, the network performs exceptionally well on the training data, exhibiting minimal training loss, but fails to generalize to new, unseen data, resulting in poor performance metrics and larger errors during testing.

1.3.5 Layers

Neural networks are composed of layers, each serving a specific purpose in the process of transforming input data into a desired output. The layers used in our BoMN model are:

Convolution layer Most popularly used with convolutional neural networks [5] this layer plays an important role in the network's functionality. It is based on working with matrices called kernels. The values in the kernel are learnable, which means they are adjusted over the training process to enhance performance. In this thesis, we will be using these hyperparameters:

- **Depth** determines the dimension of the output volume (activation maps). Influences pattern recognition as well as the number of neurons.
- **Size** determines dimensions of kernel.
- **Activation function**

The algorithm consists of a sliding kernel along the input. At every position, it calculates the sum of the element-wise multiplication of corresponding pixels in the input and kernel. The result is then inserted into the output. This process is then repeated over the whole input multiple times (depending upon the number of kernels) The result of this operation captures local patterns while preserving positional relationships.

Max Pooling layer Max pooling is an operation of non-linear down-sampling. This means that the output image of this layer is usually smaller than the input. This helps to reduce parameters for the next convolutional layer, providing faster training. This layer is defined by two hyperparameters:

- **Filter size** determines the dimensions. In the case of the filter reaching out of the array, only valid values are taken into consideration.

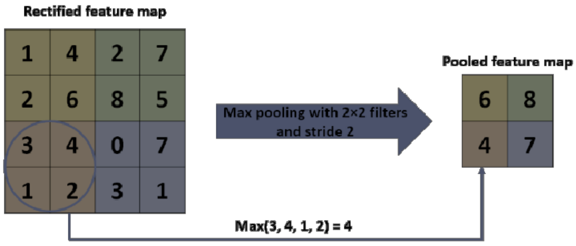

Figure 1.1: Max Pooling Example [6]

- **Stride** determines how many columns will the filter move.

The higher these hyperparameters' values are the smaller will the output be. This layer iterates over the input field and looks at the subfield with size of filter size. In this subfield, it finds the largest number and writes only the largest number into the output field. After this, the filter moves by stride columns left until all columns are checked. In that case, the filter moves back to the first column of the input field and then moves down by the stride (refer to 1.1 as an example).
This process is repeated until the whole input field is iterated.

Flatten layer The flatten layer serves as a bridge between convolutional or pooling layers and dense (fully connected) layers. Convolutional layers and pooling layers often output multi-dimensional tensors, which are unsuitable for direct input into dense layers. The flatten layer transforms these multi-dimensional tensors into a one-dimensional vector, preserving the spatial information. It reduces the dimensions of the input, creating a vector while maintaining the relationship between the data points.

Dense layer Dense layers, also known as fully connected layers, are fundamental components of neural networks. In these layers, each neuron is connected to every neuron in the previous layer, enabling feature learning and pattern recognition. Each neuron in a dense layer performs a linear transformation on the input data, followed by the application of an activation function. This process allows the network to learn and combine features from the previous layers, helping the network to understand complex relationships and interactions between them. Dense layers often produce the output, whether it's class probabilities in classification tasks or continuous values in regression tasks.

The output of a dense layer can be mathematically represented by the equation $y = f(Wx + b)$, where x is the input vector, W is the weight matrix, b is the bias vector, and f is the activation function applied element-wise. This equation describes how the dense layer transforms the input data by applying weights and biases and then passing the result through an activation function, which introduces non-linearity into the model.

Activation functions in dense layers are crucial because they enable the network to learn more complex patterns and relationships. Common activation functions include the Rectified Linear Unit (ReLU) [7], which helps the network handle non-linear relationships while mitigating the vanishing gradient problem,

Chapter 2

Related Work

The need of knowing human measurements is always present. The methods used for obtaining have been developing and nowadays are competing with traditional measurement methods in precision. The progress has been developing also thanks to new ways of obtaining data used for these estimations. First, only numerical values were available to researchers. Available measurements were used for statistical models which were used to roughly estimate average measurements.

2.1 History

Quetelet's concept of the average man has had a profound influence on the development of psychology and the statistical study of human characteristics [8]. Quetelet argued that measurements of human traits would conform to the normal distribution, with the average representing the true or ideal type. This notion of the average as representative allowed early psychologists to blur the distinction between individual-level data and aggregate statistics. The idea of the average man as a statistical model for understanding human nature persisted in psychology, even as reporting practices shifted away from individual-level results towards aggregate statistics. Quetelet's work laid the groundwork for the widespread use of large-scale data collection and analysis techniques, which became central to the field's pursuit of understanding individual differences and population-level phenomena. The legacy of Quetelet's work highlights the important epistemic challenges that arise when connecting statistical models to claims about individuals and human nature. This historical context is relevant for understanding the development of methods for estimating human body measurements from data, which often rely on aggregate statistics and population-level modeling.

Since the 18th century, the military began employing anthropometric measurements, primarily focusing on stature, to identify suitable candidates [9]. ...

2.2 Pose Estimation

Pose estimation is a computer vision task that involves detecting and locating key points on objects or persons in images or videos. These key points correspond to specific body parts or landmarks, such as joints. The goal is to accurately determine the spatial position and orientation (pose) of the person within the image or video frame. Pose estimation has various applications, including human-computer interaction, gesture recognition, action recognition, augmented reality, and robotics. It's a fundamental technology that enables machines to understand and interact with the physical world more intuitively.

2.3 Domain Adaptation

When training a machine learning model, such as a neural network, it is crucial for the performance of the model that the training and testing data are similar and follow the same distribution [10]. However, this condition is often not met. One reason for this is the limited amount of training data, as demonstrated in this thesis. Another reason is the slight differences between the test data and the data the network was trained on. For instance, this can occur with medical devices where output devices may have varying colour representations. Considering factors like time complexity and data availability, domain adaptation can be an optimal solution to address these discrepancies.

Domain adaptation is a type of transfer learning, used to mitigate domain shift between source (training data) and target (data used for testing) domain. The assumption behind domain adaptation is, that there exists a difference in domains (source and target), but not in the task. Furthermore, the domains should not be too different and should have similar probability distributions. These limitations impact the performance, up to point where network's performance may be worse than before applying domain adaptation if used incorrectly. On the other hand, there are multiple advantages to using domain adaptation. For example, the cost of collecting and labelling new data can be mitigated. In addition, the resulting network is able to generalise better on the target domain. Multiple domain adaptation approaches have been developed to address this issue:

2.3.1 Instance-Based Methods

Instance-based methods reweigh or select instances from the source domain to better match the target domain's distribution. One of the notable approaches, Kernel Mean Matching [11] (or KMM for short), aims to adjust the distribution of the source domain by assigning weights to source instances. This process minimizes the difference between the source and target distributions in a high-dimensional feature space.

2.3.2 Feature-Based Methods

Feature-based methods focus on learning a common feature representation that is invariant across domains. These methods typically involve transforming the feature space such that the source and target domains become indistinguishable. One important method is the Maximum Mean Discrepancy (MMD). DAN [12] incorporates MMD into a deep neural network, aligning the distributions of source and target features at multiple layers of the network.

Another influential method is Domain-Adversarial Neural Networks (DANN) [13]. DANN employs adversarial training, where a domain classifier is trained to be able to distinguish source and target features while the feature extractor simultaneously learns to confuse the domain classifier. This adversarial process influences the feature extractor to generate domain-invariant features, enhancing model performance on the target domain.

2.3.3 Parameter-Based Methods

Parameter-based methods involve sharing or regularizing model parameters between the source and target domains. These methods often adapt the source model to the target domain by imposing regularization constraints. Deep Domain Confusion (DDC) [14] combines a domain confusion loss with the standard classification loss. The domain confusion loss encourages the network to learn features that are not only discriminative for the source task but also invariant across domains.

2.3.4 Hybrid Methods

Hybrid methods integrate multiple adaptation strategies to leverage their complementary strengths. CyCADA (Cycle-Consistent Adversarial Domain Adaptation) [15] combines adversarial training, cycle-consistency, and feature alignment. CyCADA uses generative adversarial networks (GANs) to translate images from the source domain to the target domain and vice versa, ensuring that the translations are cycle-consistent. This approach helps align the feature distributions while preserving the semantic content, leading to improved adaptation performance.

CyCADA uses principles proposed by CycleGAN [**?**]. Firstly, it employs a generators and discriminators to create cycle consistency. The generators are used to map images from the source domain to the target domain and then back to the original domain.

2.4 Data synthesis

Neural networks have caused a shift in task solving in numerous fields. These tasks can be rather complex and require a large quantity of quality prepared datasets. To create such dataset, the data has to be collected and further correctly labelled. This copious task is however not so simple. There are many obstacles that have to be overcome to create usable dataset, of which some are:

Cost Creating datasets can be expensive as they have to be large, often requiring multiple workers to collect and label data. In some cases special devices and environment may be required to be able to obtain such data, which further increases the cost.

Time complexity As the type of data vary from task to task, the data may be complicated to obtain, such as taking photographs in different regions or measuring values using complex machinery or many others. The data collection is, however, only half of the task. The other half consists of correctly labelling the data. Depending upon the task, it can sometimes require professionals to assess the data.

Privacy In many medical applications the task requires personal data to make correct predictions. This raises concerns about private information leaking and thus the data has to be dealt with in way it follows all privacy regulations. Moreover, there must be enough willing patients to be able to procure such dataset, which can raise the monetary cost of such dataset. Medicine is not the only field in which personal information may be required to create suitable dataset.

Robustness In order for the network to be able to correctly generalise, the dataset should be as general as is possible, including extreme and unusual cases as well. If this is not the case, the network may struggle with such cases or can develop a bias to the most common example. This is often the most complicated task, and it is rarely able to fully contain the whole spectrum.

Data availability Not all data is available all the time and thus obtaining can be impossible at that specific time. Most commonly this issue rises when the subject of the dataset is rare (for example certain illnesses or animals etc.) or is currently not available (wrong season or not appropriate conditions).

2.4.1 Synthetic data

To counteract these obstacles, some data can be created synthetically. This type of data is artificially generated instead of collected manually. The generation is based on models and algorithms which aim to create samples that are adjusted to meet specific requirements, but also contain features and variation the real data may be missing. The main aim of this data is to make the network more robust, helping it to be precise even in situations that were not included in the real data [16]. Other than better precision, generating data can be far less expensive than collecting it.

There are currently multiple approaches that are used for generation of synthetic data [17]. We will present some methods that could be used with issue

2.5 Neural Anthropometer

An important article is the [18] which proposes a method to tackle this task. Its Neural Anthropometer provides a valuable approach which will we use as backbone our convolutional neural network architecture. We do not need everything used in this article as we already have annotated synthetic dataset provided by [19]. To keep the network as small as possible due to resource consumption and training difficulty increase with size. The proposed architecture starts with a binary image silhouette input. This is then processed by a convolutional layer. Number of channels was based on number of values on output. The output tensor is then passed through ReLU [7] along with batch optimization. Subsequently, a max pooling layer is used followed by a convolutional layer. The output is then once again processed by max pooling layer with configuration same as before. The result is then flattened to a tensor which is passed to a fully connected layer and a ReLU. As the last layer a regressor is used to provide the measurements estimation.

This approach is further implemented by [19]. The article delves into different data representations of human bodies and influence this

Chapter 3

Proposed solution

3.1 Problem specification

The goal of this thesis is to expand on body measurements estimation with the use of neural networks. Mainly to explore different data augmentation methods to provide better results when training is done on synthetic dataset.

3.2 Obstacles

One of the issues when working with human body measurements is the lack of real world data. The process of measuring is time-consuming and requires privacy measures to take place to protect subjects' personal information. This can be avoided by using synthetic datasets.
Moreover the time complexity to train a neural network can be reduced by using powerful device which is not available.

In this section, we'll explore the datasets utilized within this thesis. Our focus will be on 2D front-facing and profile human binary silhouettes. This form was chosen upon the data provided by the BodyM dataset. The subjects are positioned in an a-pose, ensuring greater consistency in the samples.

3.3 Datasets

3.3.1 SURREACT

Description

SURREACT [20] is a synthetic dataset built on SMPL model. The main goal of the work was to explore benefits of using synthetic data for human action recognition. The study aimed to answer whether the synthetic data could potentially improve accuracy of already existing methods. This theory was confirmed and even shown improvements over other state-of-the-art action recognition methods. This is however not as important for this thesis as we are not going to use the features that were added.

The dataset introduced by [19] is an extension of the SURREACT dataset, incorporating the data generation techniques and a custom annotation method. This thesis utilizes a modified version of this dataset. The original dataset comprises 50,000 human scans, meshes, annotations, and other data of subjects in the T-Pose. In contrast, our customized version offers 79,999 frontal and 79,999 lateral images with annotations, featuring subjects in the A-Pose. They are saved in RGBA format with dimensions of 320x240 without background thus eliminating the need of segmentation. Measurements are saved in .npy file format requiring us to use NumPy [21] to read these values.

3.3.2 BodyM

Description

This public body measurement dataset [2] contains measurement and image data from real human subjects. The subjects were photographed in a well-lit indoor setup, resulting in the data being less prone to segmentation inaccuracies. Subjects also wore tight-fitting clothing to better reflect the measurements. After the pictures were taken, the subjects were scanned using Treedy photogrammetric scanner and fitted to the SMPL mesh. Measurements were then taken on said meshes. It also promises a wide ethnicity distribution

Measurements

Information provided by BodyM dataset paper are following:

Table 3.1: Definition of annotated anthropometric body measurements. Note that the 3D model is expected to capture the human body in the default T-pose, with Y-axis representing the vertical axis, and Z-axis pointing towards the camera [19].

Body measurement	Definition
Head circumference	circumference taken on the Y-axis at the level in the middle between the head skeleton joint and the top of the head (the intersection plane is slightly rotated along X-axis to match the natural head posture)
Neck circumference	circumference taken at the Y-axis level in 1/3 distance between the neck joint and the head joint (the intersection plane is slightly rotated along X-axis to match the natural posture)
Shoulder-to-shoulder	distance between left and right shoulder skeleton joint
Arm span	distance between the left and right fingertip in T-pose (the X-axis range of the model)
Shoulder-to-wrist	distance between the shoulder and the wrist joint (sleeve length)
Torso length	distance between the neck and the pelvis joint
Bicep circumference	circumference taken using an intersection plane which normal is perpendicular to X-axis, at the X coordinate in the middle between the shoulder and the elbow joint
Wrist circumference	circumference taken using an intersection plane which normal is perpendicular to X-axis, at the X coordinate of the wrist joint
Chest circumference	circumference taken at the Y-axis level of the maximal intersection of a model and the mesh signature within the chest region, constrained by axilla and the chest (upper spine) joint
Waist circumference	circumference taken at the Y-axis level of the minimal intersection of a model and the mesh signature within the waist region – around the natural waist line (mid-spine joint); the region is scaled relative to the model stature
Pelvis circumference	circumference taken at the Y-axis level of the maximal intersection of a model and the mesh signature within the pelvis region, constrained by the pelvis joint and hip joint
Leg length	distance between the pelvis and ankle joint
Inner leg length	distance between the crotch and the ankle joint (crotch height); while the Y coordinate being incremented, the crotch is detected in the first iteration after having a single intersection with the mesh signature, instead of two distinct intersections (the first intersection above legs)
Thigh circumference	circumference taken at the Y-axis level in the middle between the hip and the knee joint
Knee circumference	circumference taken at the Y coordinate of the knee joint
Calf length	distance between the knee joint and the ankle joint

Body measurement
Ankle girth
Arm-length
Bicep girth
Calf girth
Chest girth
Forearm girth
Head-to-heel length
hip girth
leg-length
shoulder-breadth
shoulder-to-crotch length
thigh girth
waist girth
wrist girth

Issues

The paper has not provided us with any information regarding the measurement location. This requires us to believe that the measurements were taken in accordance to the norm.

Chapter 4

Implementation

4.1 Used software

Keras

Keras [22] is an open-source neural network library written in Python. Thanks to its user-friendly interface and modular design is Keras one of the leading frameworks in neural network development. Its simple yet flexible architecture allows for easy prototyping and experimentation, making it an ideal choice for both beginners and experienced practitioners in the field of deep learning.

OpenCV

Open Source Computer Vision Library (OpenCV for short) [23] is a comprehensive open-source library originally developed by Intel. It is mainly used for various tasks in fields such as computer vision or machine learning. At the time of writing this thesis, OpenCV provides over 2500 optimized algorithms. These can effectively perform many tasks such as face detection, object tracking, image preprocessing and many more. Providing interfaces in multiple programming languages such as Python, C++, Java and MATLAB it is very popular with the community as well as recognisable and famous companies.

Chapter 5

Results

Chapter 6

Conclusion

Bibliography

[1] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, pp. 248:1–248:16, Oct. 2015.

[2] N. Ruiz, M. Bellver, T. Bolkart, A. Arora, M. C. Lin, J. Romero, and R. Bala, "Human body measurement estimation with adversarial augmentation," 2022.

[3] A. Haleem, M. Javaid, R. P. Singh, S. Rab, R. Suman, L. Kumar, and I. H. Khan, "Exploring the potential of 3d scanning in industry 4.0: An overview," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 161–171, 2022.

[4] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, "Loss functions and metrics in deep learning," 2023.

[5] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[6] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," 2020.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[8] D. Tafreshi, "Adolphe quetelet and the legacy of the "average man" in psychology.," *History of Psychology*, vol. 25, no. 1, p. 34, 2022.

[9] S. Ulijaszek and J. Komlos, "12 from a history of anthropometry to anthropometric history," *Human variation: From the laboratory to the field*, p. 183, 2010.

[10] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

[11] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), vol. 19, MIT Press, 2006.

[12] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*, pp. 97–105, PMLR, 2015.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1180–1189, PMLR, 07–09 Jul 2015.

[14] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[15] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*, pp. 1989–1998, Pmlr, 2018.

[16] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," *CoRR*, vol. abs/1804.06516, 2018.

[17] Y. Lu, M. Shen, H. Wang, X. Wang, C. van Rechem, T. Fu, and W. Wei, "Machine learning for synthetic data generation: A review," 2024.

[18] Y. G. Tejeda and H. A. Mayer, "A neural anthropometer learning from body dimensions computed on human 3d meshes," *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, 2021.

[19] D. Skorvankova, A. Riečický, and M. Madaras, "Automatic estimation of anthropometric human body measurements," 12 2021.

[20] G. Varol, I. Laptev, C. Schmid, and A. Zisserman, "Synthetic humans for action recognition from unseen viewpoints," in *IJCV*, 2021.

[21] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke,

and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[22] F. Chollet *et al.*, “Keras.” `https://github.com/keras-team/keras`, 2015.

[23] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.