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


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


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 BodyM dataset [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 [4]. 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 Loss Functions

The training process is guided by the loss function [5]. 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.

### 1.3.2 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.

### 1.3.3 Backpropagation

Backpropagation [6], used as a short for backward propagation of errors, is an algorithm for training feedforward neural networks. In this type of networks the data flows in only one direction. The core idea is to minimize the error at the output layer by adjusting the weights of the network using gradient descent. The process involves two main steps:

1. **Forward Pass:** The input data are passed through every layer sequentially and all neurons process the data they have received. Resulting values are called

activations. After calculation, each neuron sends this activation multiplied by weight to all output neurons. Each neuron in the next layer then sums the activations it has received and adds a bias value. After all layers have processed the data, we are presented with the result, which is the goal of this pass.

2. **Backward Pass:** After obtaining the result from forward pass, we can now compare it with the desired result. The difference between the predicted output and the expected output is calculated using a loss function. The error indicates how imprecise the predictions are from the actual values. Calculate how much each weight in the network contributed to the error, which is done by determining the gradient of the error with respect to each weight, providing the direction and magnitude of change needed. After this, the process is propagated backwards to its preceding layer and repeated. The highest error and most influential weights are prioritised in the update process to adjust the most significant error sources first.

This process needs to be repeated for every sample in our training. To retain all these weights updates in larger models, we would require a significant amount of memory and processing power to recalculate. Instead of this, a newer approach - Stochastic Backpropagation [**?**] is used. By proving identities, we can use smaller, randomly selected batches for the training process, allowing us to use backpropagation in larger models.

**Gradient Descent** is a popular optimization algorithm used commonly with neural networks [7]. It is used for finding values of parameters of a function which help to reduce the loss by the largest margin possible [8]. The performance is gained by taking steps in the opposite direction of current gradient as this is considered to be the steepest direction for descent. There are multiple variations of gradient descent to provide better efficiency in specific situations.

### 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 [9] 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.
- **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.

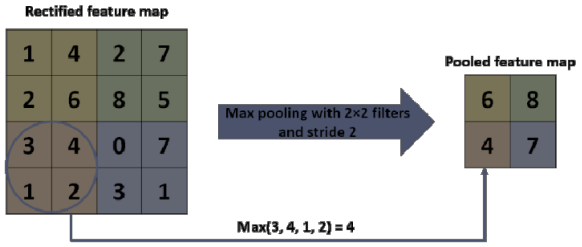


Figure 1.1: Max Pooling Example [10]

**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) [11], which helps the network handle non-linear relationships while mitigating the vanishing gradient problem,

# Chapter 2

# Related Work

The need to understand human measurements has always been essential. The methods for obtaining these measurements have continually evolved, and today, they rival traditional techniques in precision. This progress has been driven by innovative ways of collecting data for these estimations.

## 2.1 History of Anthropometry

Quetelet's concept of the "average man" [12] has profoundly influenced the development of psychology and the statistical study of human characteristics. Quetelet argued that measurements of human traits would conform to a 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. Success of this idea cannot be diminished as one of the outcomes of this the idea Body Mass Index (BMI) is still used to this day [13]. The idea of the average man as a statistical model for understanding human nature persisted in psychology, even as reporting practices shifted 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 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.

Another significant impact of anthropometry on human society is illustrated by the Bertillon system [14]. In the late 1870s, the French policeman Alphonse Bertillon began measuring prisoners for later identification. This approach helped in identifying

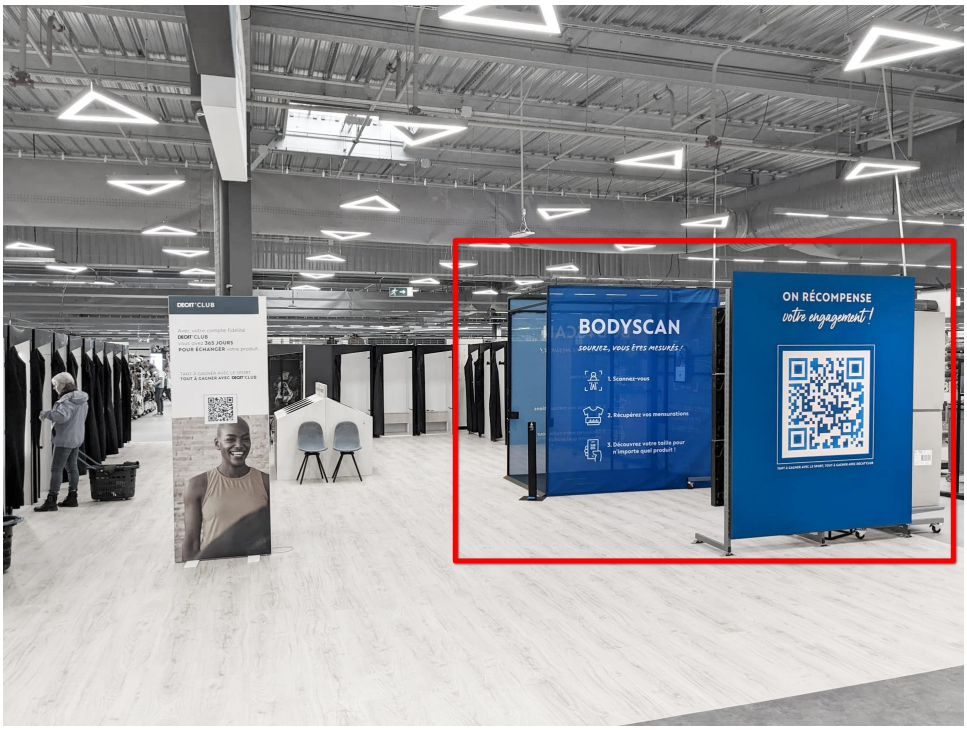


Figure 2.1: Treedy's body scanning station used in Decathlon stores. Customers can obtain their body measurements just by walking in. The device scans them using four camera-equipped devices and, using their patented Nakednet AI, estimates the measurements. Credit: Treedy's [17].

recidivists and provided the police with additional information.

In the early 20th century, anthropometric measurements took on a new function. Predictions of body fat were based on measurements such as body length, circumference, and skinfold thickness [15]. This method spread rapidly due to its non-invasiveness, portability, and cost-effectiveness.

Nowadays, anthropometric measurements remain important in multiple fields. Digital anthropometry is being used to measure the size of infants' bodies to check for nutrition and growth anomalies [16]. Additionally, companies like Treedy's [17] are incorporating digital measuring stations in clothing stores to help customers pick the correct clothing size and even recommend fitting pieces.

## 2.2 Types of Human Body Estimations

### 2.2.1 Mathematical Models

Calculating the Body Mass Index (BMI) from body dimensions [18] became crucial in tracking the development of obesity pandemic during the 20th century [13]. This led researchers to develop statistical models to detect overweight or underweight con-

ditions in the population. The measurements required for these tasks varied, but the popular approach resulted in requiring only height and weight [19] based on the BMI formula [20].

Another significant task involved determining the centre of mass of the human body, the moment of inertia, and related parameters, including BMI. These measurements are essential for various applications, such as Self-Maneuvering Units [21] used for extravehicular movement in space or the development of prosthetic devices and man-machine interfaces [22]. These tasks continue to be researched today [23], highlighting the ongoing need for such studies.

In anthropology, an anatomical task arose to estimate human stature from parts of the skeleton [24]. Researchers have used various indicators, such as brain weight [25], femur size [26], and more [27], to estimate stature, aiding scientists in understanding the shape, evolution, and environmental influences on humanity.

### 2.2.2 Regression

As powerful computational devices became available to scientists, more complex models of regression analysis have enabled more precise estimations. Over the years the popularity grew not only for precision of results, but also thanks to the ubiquity and availability of large datasets [28]. In recent years it has become an essential part of statistics for students of anthropology [29].

Regression methods often build on the same hypotheses as mathematical models. Notably Trotter and Gleser [30], used regression to estimate the relationship between different bone lengths and stature, providing study based on large dataset, incorporating aging factor and yielding groundbreaking results [31].

Regression is also used in cases when it is impossible to recover all parts of a human skeleton [32] to perform traditional forensic methods. For example, hand length [33, 34, 35, 36] or foot [33, 37] can be used. Additionally, regression can predict whole-body fat mass, lean mass and trunk fat mass [38] using BMI, WC, gender and age.

### 2.2.3 Images

More recently image data have been used to estimate stature. Images can be calibrated or uncalibrated [39, 40]. While the scale of calibrated image can be calculated with some knowledge about objects in the image, the uncalibrated image is a more complex process, often requiring reference length. A method to obtain required reference sizes is proposed by BenAbdelkader and Yacoob [39], by using statistical relationship in sizes of body parts. Although the methods, often do not focus primarily on anthropometric measurements, they usually rely on using stature for different tasks.

A task related closely to task of this thesis is presented by Meunier et al. [41]. Mentioned approach is based on taking pictures from two calibrated cameras and finding landmarks on the subject. The circumferences cannot be measured directly and have to be estimated using mathematical models. A different approach suggested by Hung et al. [42], uses a single camera and a calibration square to take three pictures (front, side, back), estimating the shape from multiple images instead of using mathematical models. A more recent approach uses five images and linear regression for measurements but requires manual landmark annotation [43]. A simple and cost-effective method uses a single webcam and an A4 sheet of paper as a scale in multiple pictures of a person to estimate dimensions using the Pixel Density Method.

### 2.2.4 Virtual Representation

Another step in anthropometry was brought by usage of 3D data. These are usually collected by special devices that provide extra information over usual cameras (see Section 1.1.2 for more information). After all necessary data is captured, a processing algorithm (such as Iterative Closest Point [44]) is run to unify data from multiple images and as result generates a point cloud or a 3D mesh [45]. These structures are proficient in providing the necessary means to calculate the required measurements. The measuring process can be done manually [46], providing the user with measurements at selected points, or by automatic landmark locating. Approach suggested by Tomi et al. [47] uses angle-based algorithm to calculate the width and thickness of subject's body. Then, using joint information provided by capturing device's (Kinect) SDK, finds the leftmost and rightmost points of given measurement and calculate the distance between them. For circumferences, they use ellipse perimeter equation. Different approach uses a deep neural network to extract required measurements [48]. This approach is based upon a template on which the measuring points are defined. The network takes the template and deforms it to look like the input. The output of this network is a deformed template with measuring points, that can be measured with point-to-point distance.

## 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 [49]. 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 [49]. 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 [50].

### 2.3.1 Instance Re-weighting Methods

Instance re-weighting aims to mitigate the issue of sample selection bias by assigning different weights to source domain instances based on their relevance to the target domain distribution. Sample selection bias occurs when the samples selected for model training are not representative of the population or the conditions that the model will encounter. This bias can lead to skewed or incorrect conclusions. One of the notable approaches, Kernel Mean Matching [51] (or KMM for short), aims to adjust the distribution of the source domain by assigning weights to source instances.

### 2.3.2 Feature Adaptation Methods

Feature-based methods focus on finding 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 [52] 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) [53]. 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 Classifier Adaptation Methods

Classifier adaptation methods aim to develop a general classifier by utilizing labeled samples from the source domain along with a few labeled samples from the target domain.

ASVM [54] is an example of kernel classifier-based methods. It adjusts the decision boundary learned from the source domain to fit the target domain by adding a bias function $\Delta f(x)$ to the source classifier. This bias function is parameterized by $w$ and optimized using SVM techniques to minimize classification errors on the target domain data.

### 2.3.4 Deep Network Adaptation Methods

Deep network adaptation is a prominent technique in domain adaptation, leveraging the powerful feature representation and end-to-end training capabilities of deep neural networks. The methods mentioned so far can be used in deep neural networks

### 2.3.5 Adversarial Adaptation Methods

This approach focuses on minimizing domain discrepancies by training models to generate domain-invariant features or pixel-level target samples. CyCADA (Cycle-Consistent Adversarial Domain Adaptation) [55] combines adversarial training, cycle-consistency, and feature alignment. CyCADA uses generative adversarial networks (GANs) to generate images from domains. This approach helps align the feature distributions while preserving the semantic content, leading to improved adaptation performance.

CyCADA uses principles proposed by CycleGAN [56]. 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. The difference between input image and the image transformed back to source domain is used as a loss called cycle-consistency loss. The whole loss is then created by summing the cycle consistency loss and two generator losses.

## 2.4 Data Acquisition

Neural networks have caused a shift in task solving in numerous fields. The tasks, which can be neural networks used for are rather complex and require a large amount of carefully prepared datasets. To create such dataset, the data has to be collected and further correctly labelled. There are, however, 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** 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.

**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 overcome these obstacles, some data can be created synthetically [57]. This type of data is artificially generated instead of being collected manually. The generation is based on models and algorithms designed to create samples that meet specific requirements while also containing features and variations that real data may lack. The primary goal of synthetic data is to make the network more robust, enabling it to be precise even in scenarios not covered by the real data [58]. Besides improved precision, generating data can be significantly less expensive than collecting it. Data can be generated from existing real data, or by using domain-specific models, or by combining both methods [59].

There are currently multiple approaches that are used for generating synthetic data. The simpler methods involve basic image manipulations [60]. Geometric transformations, such as rotation, scaling, cropping and flipping have proven to be useful when applied correctly. An example of incorrect use would be flipping images of CIFAR10, which can lead to label mismatch when used with numbers 6 and 9. Another group of transformations, known as photometric methods, changes the RGB channels. Examples include color jittering (replacing colours with different random or set colours), edge enhancement, and fancy PCA [61]. Photometric augmentation have, however shown smaller improvement compared to geometric transformations [62].

A somewhat counterintuitive approach was proposed by Inoue [63] and later improved by Summers and Dinneen [64], who suggested mixing parts of images. This augmentation surprisingly yielded better results than the base model. Another interesting method involves removal of random segments of an image [65] and replacing the areas with either gray or random values, with the latter one bringing better improvement. This method is also valuable in dealing with overfitting. Both of these methods must be used carefully to ensure they preserve the labels of the dataset.

With rise of deep learning, new methods of generating synthetic data have emerged. One of these is the Synthetic Minority Over-Sampling Technique (SMOTE [66]), which is a widely used technique to address class imbalance issues. This method combines $k$ nearest neighbours to create new instances. Thanks to its popularity, many alternations of this method have been developed [67, 68, 69].

Another popular method involves using generative adversarial networks (GANs). These networks consist of two modules - discriminator and a generator. The job of the generator is to create images that resemble those in the dataset. On the other hand the job of the discriminator is to correctly determinate, whether an image is from the dataset or generated. When the discriminator assumes correctly, the generator improves, generating harder-to-distinguish images. In case the discriminator is wrong, it imporoves its ability to discern the images. This iterative learning process is then iterated over many iterations, resulting in generator capable of generating images almost indiscernible from the original dataset. One of the most popular implementations is the CycleGAN [56] (see Section 2.3.5). A different method was proposed by Karras et al. [70], which begins its training low-resolution images and gradually increases the resolution by adding layers. This approach's advantage is that it first discovers the large-scale structure of the images and then focuses on finer details as training progresses. To prevent shocks from new layers being added, the method proposes smoothly fading the new layers in.

## 2.5 BMnet

BMnet [2] is a network based on MnasNet [71], which is a convolutional neural network. As an input it requires a frontal silhouette and optionally works with lateral silhouette as well. The lateral silhouettes provide additional information for chest and waist measurements. Moreover, the paper uses height and weight as an input metadata. The height is necessary in situations where the subjects have variable distance to camera making the scale of the subjects non-uniform. Weight then provides yet another information of the subject's size and shape. In the network the images are concatenated spatially, whilst the height and weight are concatenated depth-wise. This object is then evaluated by the MnasNet followed by MLP, consisting of 128 neurons and 14 outputs, providing the results. The adversarial body simulation (ABS) presented in BMnet work with SMPL [1] model (see Section 1.2) . Its main goal is to find body shapes that are challenging for BMnet, since they are not as common or even non-existent in the dataset.

## 2.6 Neural Anthropometer

An important article is by Tejeda and Mayer [72] which proposes a method to tackle the task of estimating anthropometric measurements. The Neural Anthropometer provides a valuable approach. 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.

This approach is further implemented by [73]. The article further delves into different data representations of human bodies and their performance.

# Chapter 3

# Solution

## 3.1 Proposed solution

In this section we will provide an overview of assets used in this thesis. The aim of the thesis is to explore the task of body measurements estimation. Using state-of-the-art deep learning methods we aim to evaluate usage of synthetic data compared to real human body data. This should provide valuable information about domain gap and advantages of augmenting real data with synthetic images for training purposes. In this thesis we have trained multiple networks (see section 3.1.3) on different datasets (see section 3.1.2).

### 3.1.1 Obstacles

One significant challenge in working with human body measurements is the lack of real-world data. The measurement process is time-consuming and requires extensive privacy measures to protect subjects' personal information. This issue can be addressed by using synthetic datasets.

Another challenge is the complexity of the task, which requires larger networks with numerous training parameters. Training these networks efficiently requires substantial computing power. Unfortunately, we did not have access to such devices, resulting in single training sessions taking multiple days.

### 3.1.2 Datasets

In this section, we will examine the datasets used in this thesis. Our focus is on 2D frontal and lateral human binary silhouettes. This data structure was selected based on the data provided by the BodyM dataset. The subjects are positioned in an anatomical pose [74], ensuring consistency and stability in their postures.

### SURREACT

SURREACT [75] 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 created by Škorvánková et al. [73] 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 - Surreact-APose. 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 anatomical 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, providing speed in loading when compared to using .txt or .csv files. To read them, we are required to use NumPy [76].

Measurements provided by this dataset can be found in table 3.1

### BodyM

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

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.

Table 3.1: Definition of annotated anthropometric body measurements. Note that the 3D model is expected to capture the human body in anatomical pose, with Y-axis representing the vertical axis, and Z-axis pointing towards the camera [73].

| 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 |

Table 3.2: The mean values of measurements in the Surreact-APose dataset. Averages are stated in cm and rounded to two decimal places

| Body measurement | Mean |
|---|---|
| Head circumference | 60.57 |
| Neck circumference | 36.35 |
| Shoulder-to-shoulder | 35.87 |
| Arm span | 175.35 |
| Shoulder-to-wrist | 51.11 |
| Torso length | 50.90 |
| Bicep circumference | 29.88 |
| Wrist circumference | 17.16 |
| Chest circumference | 99.71 |
| Waist circumference | 88.39 |
| Pelvis circumference | 104.81 |
| Leg length | 78.05 |
| Inner leg length | 72.87 |
| Thigh circumference | 52.31 |
| Knee circumference | 37.74 |
| Calf length | 40.39 |

Table 3.3: The mean values of measurements in the BodyM dataset. Averages are stated in cm and rounded to two decimal places

| Body measurement | Mean |
|---|---|
| Ankle girth | 24.1 |
| Arm-length | 49.43 |
| Bicep girth | 30.28 |
| Calf girth | 37.23 |
| Chest girth | 101.42 |
| Forearm girth | 26.38 |
| Head-to-heel length | 171.61 |
| hip girth | 102.21 |
| leg-length | 78.1 |
| shoulder-breadth | 35.65 |
| shoulder-to-crotch length | 64.65 |
| thigh girth | 53.83 |
| waist girth | 89.26 |
| wrist girth | 16.63 |

### 3.1.3 Neural Networks

#### Conv-BoDiEs

#### ResNet50V2

#### CenterNet

### 3.1.4 Used software

#### Keras

Keras [77] 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) [78] 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.

## 3.2 Implementation

# Chapter 4

# Results

# Chapter 5

# 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. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," *Pattern recognition*, vol. 43, no. 8, pp. 2666–2680, 2010.

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

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986," *Biometrika*, vol. 71, pp. 599–607, 1986.

[7] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[8] S. H. Haji and A. M. Abdulazeez, "Comparison of optimization techniques based on gradient descent algorithm: A review," *PalArch's Journal of Archaeology of Egypt/Egyptology*, vol. 18, no. 4, pp. 2715–2743, 2021.

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

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

[11] 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.

[12] G. Eknoyan, "Adolphe quetelet (1796–1874)—the average man and indices of obesity," 2008.

[13] G. A. Bray, "Beyond bmi," *Nutrients*, vol. 15, no. 10, p. 2254, 2023.

[14] *Suspect identities: A history of fingerprinting and criminal identification.*

[15] J. Wang, J. Thornton, S. Kolesnik, and R. Pierson Jr, "Anthropometry in body composition: an overview," *Annals of the New York Academy of Sciences*, vol. 904, no. 1, pp. 317–326, 2000.

[16] A. Ardhiansyah and A. Nasution, "Pose-independent digital anthropometry based on perpendicular two-dimensional imagery," in *2023 8th International Conference on Instrumentation, Control, and Automation (ICA)*, pp. 247–252, 2023.

[17] "Treedy's website."

[18] M. L. Pollock, E. E. Laughridge, B. Coleman, A. Linnerud, and A. Jackson, "Prediction of body density in young and middle-aged women," *Journal of Applied Physiology*, vol. 38, no. 4, pp. 745–749, 1975.

[19] D. D. Bois and E. D. Bois, "A height-weight formula to estimate the surface area of man," *Proceedings of the Society for Experimental Biology and Medicine*, vol. 13, no. 4, pp. 77–78, 1916.

[20] D. Bois, "A formula to estimate the approximate surface area if height and weight be known. 1916," *Nutrition*, vol. 5, p. 303, 1989.

[21] E. P. Hanavan *et al.*, *A mathematical model of the human body*, vol. 32. Aerospace Medical Research Laboratories, Aerospace Medical Division, Air ..., 1964.

[22] R. Contini, R. J. Drillis, and M. Bluestein, "Determination of body segment parameters," *Human Factors*, vol. 5, no. 5, pp. 493–504, 1963.

[23] L. Tesio and V. Rota, "The motion of body center of mass during walking: a review oriented to clinical applications," *Frontiers in neurology*, vol. 10, p. 460283, 2019.

[24] D. Thomas, "Methods of estimating the height from parts of the skeleton," *Medical Record (1866-1922)*, vol. 46, no. 10, p. 293, 1894.

[25] J. Marshall, "Relations between the weight of the brain and its parts, and the stature and mass of the body, in man," *Journal of Anatomy and Physiology*, vol. 26, no. Pt 4, p. 445, 1892.

[26] J. Beddoe, "On the stature of the older races of england, as estimated from the long bones," *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 17, pp. 201–209, 1888.

[27] K. Pearson, "Iv. mathematical contributions to the theory of evolution.—v. on the reconstruction of the stature of prehistoric races," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, no. 192, pp. 169–244, 1899.

[28] L. W. Konigsberg, S. M. Hens, L. M. Jantz, and W. L. Jungers, "Stature estimation and calibration: Bayesian and maximum likelihood perspectives in physical anthropology," *American Journal of Physical Anthropology: The Official Publication of the American Association of Physical Anthropologists*, vol. 107, no. S27, pp. 65–92, 1998.

[29] L. Madrigal, *Statistics for anthropology*. Cambridge University Press, 2012.

[30] M. Trotter and G. C. Gleser, "Estimation of stature from long bones of american whites and negroes," *American journal of physical anthropology*, vol. 10, no. 4, pp. 463–514, 1952.

[31] E. T. Brandt, "Stature wars: Which stature estimation methods are most applicable to modern populations?," 2009.

[32] R. Verma, K. Krishan, D. Rani, A. Kumar, and V. Sharma, "Stature estimation in forensic examinations using regression analysis: a likelihood ratio perspective," *Forensic Science International: Reports*, vol. 2.

[33] K. Krishan, T. Kanchan, and A. Sharma, "Multiplication factor versus regression analysis in stature estimation from hand and foot dimensions," *Journal of forensic and legal medicine*, vol. 19, no. 4, pp. 211–214, 2012.

[34] M. Vukotic, "Body height and its estimation utilizing hand length measurements in montenegrin: National survey..," *International Journal of Morphology*, vol. 40, no. 2, 2022.

[35] P. Indah, A. Sari, M. Suryoputro, and H. Purnomo, "Prediction of elderly anthropometric dimension based on age, gender, origin, and body mass index," in *IOP Conference Series: Materials Science and Engineering*, vol. 105, p. 012024, IOP Publishing, 2016.

[36] O. Miguel-Hurtado, R. Guest, S. V. Stevenage, G. J. Neil, and S. Black, "Comparing machine learning classifiers and linear/logistic regression to explore the relationship between hand dimensions and demographic characteristics," *PloS one*, vol. 11, no. 11, p. e0165521, 2016.

[37] E. T. Brandt, "Stature wars: Which stature estimation methods are most applicable to modern populations?," 2009.

[38] M. R. Salamat, A. Shanei, A. H. Salamat, M. Khoshhali, and M. Asgari, "Anthropometric predictive equations for estimating body composition," *Advanced biomedical research*, vol. 4, no. 1, p. 34, 2015.

[39] C. BenAbdelkader and Y. Yacoob, "Statistical body height estimation from a single image," in *2008 8th IEEE international conference on automatic face & gesture recognition*, pp. 1–7, IEEE, 2008.

[40] C. Barron and I. A. Kakadiaris, "Estimating anthropometry and pose from a single image," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 1, pp. 669–676, IEEE, 2000.

[41] P. Meunier and S. Yin, "Performance of a 2d image-based anthropometric measurement and clothing sizing system," *Applied ergonomics*, vol. 31, no. 5, pp. 445–451, 2000.

[42] P. C.-Y. Hung, C. P. Witana, and R. S. Goonetilleke, "Anthropometric measurements from photographic images," *Computing Systems*, vol. 29, no. 764-769, p. 3, 2004.

[43] Y. Liu, A. Sowmya, and H. Khamis, "Single camera multi-view anthropometric measurement of human height and mid-upper arm circumference using linear regression," *PloS one*, vol. 13, no. 4, p. e0195600, 2018.

[44] X. Li and S. Iyengar, "On computing mapping of 3d objects: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 1–45, 2014.

[45] S. B. Heymsfield, B. Bourgeois, B. K. Ng, M. J. Sommer, X. Li, and J. A. Shepherd, "Digital anthropometry: a critical review," *European journal of clinical nutrition*, vol. 72, no. 5, pp. 680–687, 2018.

[46] R. P. Pargas, N. J. Staples, and J. S. Davis, "Automatic measurement extraction for apparel from a three-dimensional body scan," *Optics and Lasers in Engineering*, vol. 28, no. 2, pp. 157–172, 1997.

[47] B. Tomi, M. S. Sunar, F. Mohamed, T. Saitoh, M. K. B. Mokhtar, and S. M. Luis, "Dynamic body circumference measurement technique for a more realistic virtual fitting room experience," in *2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pp. 56–60, IEEE, 2018.

[48] N. N. Kaashki, P. Hu, and A. Munteanu, "Anet: A deep neural network for automatic 3d anthropometric measurement extraction," *IEEE Transactions on Multimedia*, vol. 25, pp. 831–844, 2021.

[49] 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.

[50] L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," 2020.

[51] 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.

[52] 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.

[53] 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.

[54] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proceedings of the 15th ACM international conference on Multimedia*, pp. 188–197, 2007.

[55] 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.

[56] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2020.

[57] 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.

[58] 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.

[59] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and gans," *Mathematics*, vol. 10, no. 15, p. 2733, 2022.

[60] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[61] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE symposium series on computational intelligence (SSCI)*, pp. 1542–1547, IEEE, 2018.

[62] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE symposium series on computational intelligence (SSCI)*, pp. 1542–1547, IEEE, 2018.

[63] H. Inoue, "Data augmentation by pairing samples for images classification," *arXiv preprint arXiv:1801.02929*, 2018.

[64] C. Summers and M. J. Dinneen, "Improved mixed-example data augmentation," in *2019 IEEE winter conference on applications of computer vision (WACV)*, pp. 1262–1270, IEEE, 2019.

[65] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 13001–13008, 2020.

[66] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[67] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[68] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*, pp. 878–887, Springer, 2005.

[69] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," *BMC bioinformatics*, vol. 14, pp. 1–16, 2013.

[70] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[71] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," 2019.

[72] 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.

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

[74] B. Penders, R. Brecheisen, A. Gerver, G. van Zonneveld, and W.-J. Gerver, "Validating paediatric morphometrics: body proportion measurement using photogrammetric anthropometry," *Journal of pediatric endocrinology and metabolism*, vol. 28, no. 11-12, pp. 1357–1362, 2015.

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

[76] 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.

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

[78] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.