Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Deep Learning-Based Human Body Segmentation on 3D Body Scans

Bachelor Thesis

2022
Erik Róbert Ján Jakubovský

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Deep Learning-Based Human Body Segmentation on 3D Body Scans

Bachelor Thesis

Study Programme:	Applied Informatics
Field of Study:	Applied Informatics
Department:	Department of Applied Informatics
Supervisor:	Mgr. Dana Škorvánková

Bratislava, 2022
Erik Róbert Ján Jakubovský

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Erik Róbert Ján Jakubovský
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: Deep Learning-Based Human Body Segmentation on 3D Body Scans
Segmentácia ľudského tela v 3D skenoch pomocou hlbokého učenia

Anotácia: Segmentácia ľudského tela má dôležitú rolu v kontexte analýzy ľudského tela. Častokrát je využívaná ako medzikrok v procese riešenia komplexnejších úloh, ktoré vyžadujú porozumenie štruktúre ľudského tela. V súčasnosti sa prevažná väčšina výskumu v danej oblasti orientuje na metódy strojového učenia, nakoľko presnosťou výsledkov prekonávajú analytické metódy. Segmentácia tela z 3D vstupných dát môže byť prínosom v porovnaní s použitím 2D vstupných obrazov, keďže pridané hĺbkové informácie ponúkajú potenciálne zlepšenie presnosti .

Cieľ: Účelom tejto práce je využiť nástroje strojového učenia na presnú segmentáciu ľudského tela do jednotlivých oblastí tela, pomocou 3D skenov tela na vstupe. Prvým krokom je generácia správnych anotácií k reálnym 3D dátam, ktoré budú slúžiť ako referenčné požadované výstupy počas trénovania siete. Následne po prieskume v oblasti techník strojového učenia, natrénovať a otestovať vybrané modely neurónových sietí. Konečným cieľom je vyhodnotenie výsledkov a ich porovnanie so súčasnými existujúcimi metódami.

Kľúčové slová: počítačové videnie, segmentácia tela, storjové učenie, neurónové siete, mračná bodov

Vedúci: Mgr. Dana Škorvánková
Konzultant: RNDr. Martin Madaras, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 01.10.2021

Dátum schválenia: 06.10.2021 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: Erik Róbert Ján Jakubovský
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: Deep Learning-Based Human Body Segmentation on 3D Body Scans

Annotation: Human body part segmentation has an important role in the context of human body analysis. It is often used as an intermediate step in order to solve more complex tasks, which require understanding of human body structure. These days, most of the related research is oriented on machine learning methods, since they proved to outperform the analytical approaches. Performing a body segmentation on 3D input data might be beneficial in comparison to using 2D input images, offering a potential improvement in accuracy by providing the depth information.

Aim: The aim of the thesis is to use machine learning tools to accurately segment a human body into particular body regions, using 3D body scans as an input. As a preliminary step, it is necessary to correctly annotate the real-world 3D data to generate the ground truth for network training. Then, study the machine learning techniques, train and validate selected neural models. Finally, the goal is to evaluate the results and compare the performance to state-of-the-art methods.

Keywords: computer vision, body segmentation, machine learning, neural networks, point clouds

Supervisor: Mgr. Dana Škorvánková
Consultant: RNDr. Martin Madaras, PhD.
Department: FMFI.KAI - Department of Applied Informatics
Head of department: prof. Ing. Igor Farkaš, Dr.

Assigned: 01.10.2021

Approved: 06.10.2021 doc. RNDr. Damas Gruska, PhD.
Guarantor of Study Programme

Student

Supervisor

Acknowledgments: I would like to thank my supervisor Mgr. Dana škorvánková for her guidance and help while working on this thesis.

Abstrakt

So zvyšujúcou dostupnosťou dát sa mnohé algoritmy strojového učenia stali vhodnou možnosťou pre riešenie netriviálnych úloh vyžadujúcich automatizáciu. Jednou z takých úloh je segmentácia ľudského teľa, ktorá predstavuje neoddeliteľnú súčasť analýzy ľudského tela pre dalsie spracovanie.

Segmentáciu ľudského tela robí náročnou predovšetkým vysoký stupeň voľnosti pózy, variabilita tvaru ľudského tela a rôzne šatstvo.

Cieľom tejto práce je stanoviť efektivitu vybraných architektúr hĺbkového učenia pre úlohu segmentácie mračien bodov predstavujúcich ľudské telo. Mračná bodov sú absolútne odlišnou štruktúrou v porovnaní s 2D obrázkami, majúc svoje vlastné obmedzenia a výzvy, ktoré vyžadujú iný prístup pri segmentácii.

PointNet bola jedna z prvých architektúr, ktorá priamo spracovala mračná bodov, jej nevýhodou bola slabá schopnosť zachytiť lokálne príznaky. Tento nedostatok adresoval PoinNet++, ktorý mal ale svoje vlastné nedostatky, kvôli neefektívnym algoritmom na podvzorkovanie. Tento problém zas vyriešila architektúra RanLA-Net. V našej práci sme trénovali a testovali PointNet, PointNet++ a RandLA-Net, ktoré preukázali schopnosť segmentovať ľudské telo na regióny s relatívne vysokou presnosťou.

Kľúčové slová: počítačové videnie, segmentácia tela, storjové učenie, neurónové siete, mračná bodov

Abstract

With the increasing amount of data availability, machine learning approaches became viable option for many non-trivial task requiring automation. One of the task is human body segmentation, which is an important part of human pose analysis for further processing.

Human body segmentation proves to be difficult task, the main reasons being high degree of freedom in pose, body shape variability and different clothing.

The aim of this thesis is to assess effectivity of selected deep learning architectures for point clouds segmentation of human body. Point clouds are fundamentally different structure in comparison to 2D images, having their own limitations and challenges therefore requiring different approach in neural network segmentation.

PointNet was one of the first neural network to directly process raw point clouds, however the architecture had limitations in capturing of local structures. As a result PointNet++ was introduced to tackle this problem. PointNet++ uses inefficient subsampling methods, which were addressed and improved in RandLA-Net. In our experiment, we trained and tested PointNet, PoinNet++ and RandLA-Net, which exhibited ability to segment human body into regions with relatively high accuracy.

Keywords: computer vision, body segmentation, machine learning, neural networks, point clouds

Contents

List of Figures

List of Tables

Introduction

With the increasing amount of data availability machine learning approaches became viable option for many non-trivial task requiring automation. One such task is human body segmentation which is an important human pose analysis for further processing in fields such as human-machine interaction, surveillance, garment retexturing, motion capture, gaming etc. [17].

To us, human body figure can be easily recognised, we can easily distinguish between different limbs and people, however machines have problem grasping the intricacies of human body. Human body has high degree of freedom in pose, meaning human body can express many different poses ranging from walking, sitting to dancing, jumping or doing a handstand. Moreover no two people are the same, each person behaves differently, sports different clothing, has different hairstyle, body shape or different skin or hair colour. All this information noise and variations of human body make effective human body parsing non-trivial task.

Since automatic segmentation proves to be so difficult many proposed datasets for human body segmentation and pose estimation utilise 3D meshes with 3D skeletons to generate datasets with ground truth labels. Machine learning algorithms trained using synthetic datasets tend not to generalise well for real life situations [15]. Thus we chose to use real human body scans with manually annotated skeletons [13]. Using 3D skeleton we can automatically annotate 3D scans.

With the sudden boom in machine learning many machine learning based approaches for human body segmentation or pose estimation are being researched. Since 2D images are readily available and can be easily captured many proposed solution are image oriented [16, 25, 10]. Although 2D images capture human pose relatively well, they are prone to segmentation inaccuracies caused by lack of depth, body occlusion or parallax. Moreover 2D images contain noisy background. For this very reason we have decided to inspect other media for storing human body pose, namely unstructured point clouds. Using a 3D scanner a human body pose can be captured and stored as an unstructured point cloud. In comparison to 2D images point cloud capture depth and are resistant to body occlusion and parallax. Nonetheless, point clouds are fundamentally different structure compared to 2D images possessing their own challenges and limitations [21], therefore requiring different approach in neural network segmentation.

Point clouds are essentially sets of points possessing 3D coordinates and sometimes colour features. Unlike 2D images, they lack any inherent structure, making standard convolutional networks ill-suited.

In this thesis we aim to analyse effectivity of existing neural networks for point cloud part segmentation on 3D body scans of real humans. This work is directed towards scans of individual persons without any voxelization or other type of pre-processing.

In the following chapter we explain some basic concepts in field of machine learning and segmentation. In chapter 2, we explore work and papers of authors dealing with similar problems. The articles are split into analytical methods, machine learning methods and methods for point cloud segmentation. In chapter 3, we propose our solution, which is further elaborated in chapter 4. In chapter 5, we evaluate our results and explain used metrics. In last chapter we conclude our work.

Chapter 1

Problem Overview

In this chapter we briefly explain basics of machine learning, image segmentation and unstructured point clouds.

1.1 Machine Learning

We are living in the era of big data. Each day new data is being uploaded all around the globe. This increased data traffic is caused by digitalisation of our lives. Each day new transactions are being carried out, new videos are uploaded or new statistical data is collected. This data overload calls for automatic processing, which can be addressed by machine learning. The main aim of machine learning is to approximate a nontrivial function, which automatically detects patterns in input data and based on these patterns learns to solve the proposed problem. Essentially machine learning is a study of computer algorithms which compute specific tasks and demonstrate the ability to improve their performance automatically from experience [18]. The most common type is supervised learning, in which the program learns mapping:

$$f : x \rightarrow y$$

Where x are the features and y are the labels [19]. In order for the machine to learn the mapping f it needs examples and desired outputs which substitute the teacher. In optimal scenario the algorithm learns to generalise the mapping f to correctly label unseen features. Another type of machine learning is unsupervised learning. The trained model lacks any ground truth and tries to cluster similar data based on knowledge from prior experience. The last one of the three basic machine learning approaches is reinforcement learning. Similarly this approach does not use any ground truth, instead interpreter rewards agent (model) for an action taken in environment. In all three approaches a loss function is calculated to calculate how well the machine learning algorithm performs. Then parameters of machine learning algorithms are readjusted to lower the loss function in future predictions.

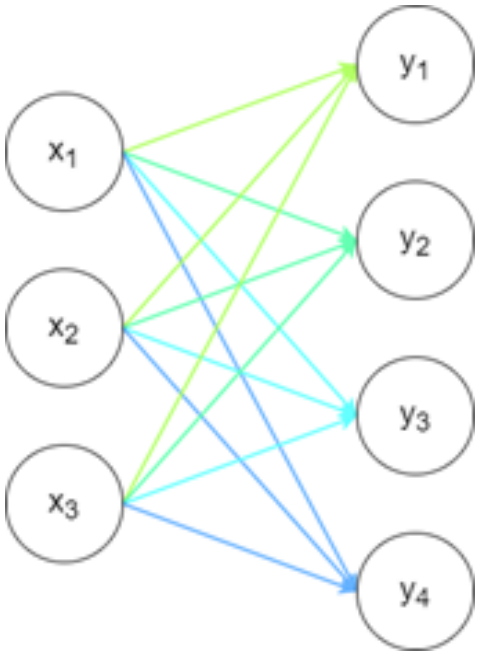

Figure 1.1: Scheme of a fully connected layer.

1.1.1 Deep Learning

Deep learning is a subfield of machine learning, which is utilises a algorithms heavily inspired by physiology of human neurons. Human neurons can be excited through dendrites at one end. When the excitation exceeds a threshold the neuron caries the signal trough the body along the axon to axon terminals, which in turn excites neurons in close proximity. This behaviour is modelled by fully connecter layers in deep learning algorithms. Fully connected layer is a type of layer in neural networks where each component of input signal contributes to each component/neuron of output signal, where contribution of each input signal is defined by weights of neuron [2].

The strength of i-th component of output signal is [9]:

$$y_i = \varphi\left(\sum_{j=1}^{m} w_i x_j + b\right)$$

Where x_j is j-th element of input signal x, w_i is weight of i-th neuron, b is bias and ϕ is non-linearity function. Using non-linearity enhances the expressiveness of layer. Goal of MLP is to approximate some function y = f(x), which is achieved by learning parameters W in mapping y = g(x, W) [11]. MLP is simply a neural network consisting of multiple fully-connected layers stacked on top of each other [2, 20]:

If output y of single fully connected layer can be described as:

$$y = \varphi_1\left(W_1 x + b_1\right)$$

Then output of n-layer Neural Network is:

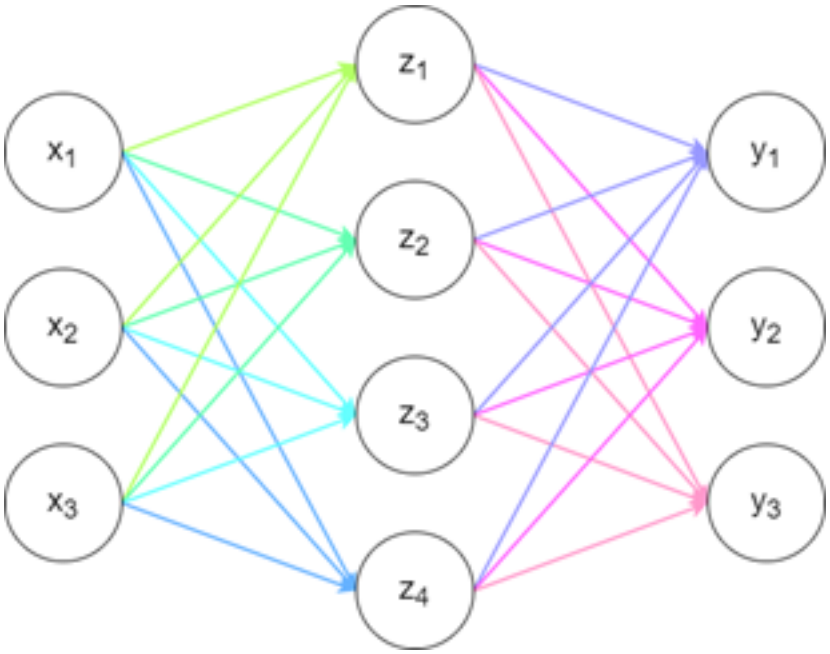

Figure 1.2: MLP consisting of one input layer, one hidden layer and one output layer (2-layer neural network).

$$y = W_n \varphi_n \left(W_{n-1} \varphi_{n-1} \left(\ldots \right) + b_{n-1} \right) + b_n$$

Where W_n is weight matrix of layer n, ϕ_n is the non-linearity function of layer n and b_n is bias of layer n. Without the non-liner functions MLP would simply express linear function which is not desired when the approximated function is nonlinear, therefore activation/non-linear functions are used in-between layers.

1.1.2 Convolutional Neural Networks

Since regular MLPs do not scale well in image classification due to large number of parameters, a different approach has to be taken. This problem is tackled by convolutional neural networks. Three types of layers are used in CNNs, namely: convolutional, pooling and fully-connected. Convolutional layer consists of multiple filters of size m x n which are convolved across the width and height of input volume producing activation map [1]. The operation of convolution can be summarised as [3]:

$$g\left(x, y\right) = w \times f\left(x, y\right) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} w\left(dx, dy\right) f\left(x + dx, y + dy\right)$$

Where f(x,y) is the function of input signal w is the kernel of dimensions 2a x 2b. Therefore the element in i-th row and j-th column of activation map y can be computed as [20]:

$$y_{ij} = \varphi\left(b + \sum_{k=0}^{m}\sum_{l=0}^{n} w_{kl} x_{i+k,\ k+l}\right)$$

Oftentimes to reduce the size of activation maps pooling layers are used, most commonly max pooling, which outputs maximum activation in specified region [1]. Generally for input size of $H_1W_1D_1$ and stride S and spatial extent F the output has dimensions $H_2W_2D_2$ where [1]:

$$H_2 = \frac{(H_1 - F)}{S} + 1$$
$$W_2 = \frac{(W_1 - F)}{S} + 1$$
$$D_2 = D_1$$

After pooling layer the activation map is processed by fully connected layer, where each activation of map is passed into neuron, producing new feature vector.

1.2 Segmentation

Image segmentation is the practice of partitioning image into meaningful regions either group of pixels, points or polygons. Regions The two objectives of segmentation are to decompose images into parts for further processing and to change the representation of images where the parts are more meaningful or more efficient for processing [14]. Image can be segmented into regions by finding pixels representing borders, by clustering of pixels or by separating image into foreground or background. Most of existing literature is heavily oriented toward 2D images, however the term is widely used in segmentation of different media such as point clouds, 3D meshes or depth images. So more broadly speaking, segmentation is the partition of input medium into sets where each element in set shares some common characteristic. In machine learning the segmentation task can be split into sematic segmentation, instance segmentation, object detection and lastly classification and localisation. Sematic segmentation is simply classification of each pixel in image. classification and localisation consist of finding bounding box for an object and classifying the object, while object detection find binding box and classification for multiple objects. Instance segmentation recognises multiple objects and labels each. Instance segmentation recognises multiple entities of the same class.

1.3 Unstructured Point Clouds

Point clouds are structures created by 3D scanners such as Lidar or 3D laser scanners [5]. Point clouds are essentially a set of points. Each point has three spatial coordinates

and sometimes possess RGB colour information in additional three channels. Since they are unstructured the order of points does not matter. As proposed in [21] point clouds poses three distinct properties, which must be kept in mind when proposing a machine learning algorithm: points in point clouds are unordered, invariance under transformation and interaction among points. Since each point is indexed in tensor they could convey the idea of some structure. Hence the proposed solution must disregard order of points in input tensor, since they are unordered. A network must be invariant to every single permutation of points in point set. Each point cloud represents a geometric object, therefore it represents the same object even if its translated or rotated, ergo the learned representation should be invariant to some transformations. Even though points in point cloud lack any structure, they are not isolated. Points located in close proximity do in fact represent some local structure, thus a proposed architecture must capture these local structures.

Chapter 2

Related Work

This chapter lists and sums up some of the research papers dealing with human body parsing. The existing approaches are split into two chapters. Analytical methods which solve the task of human body parsing using various non machine learning based algorithms and machine learning approaches, which utilise power of deep learning. Next we list some of the existing machine learning based solutions for point cloud segmentation.

2.1 Analytical Methods for Body Parsing

There have been many different analytical approaches for human body parsing showing promising results. Hsieh et al. presented a solution for human body segmentation in video sequences via technique of deformable triangulation. Firstly, background subtraction is used to extract human pose, using Delaunay triangulation on pose outline the body is partitioned into triangular mesh. Each centroid of triangle in mesh represents a node in graph. Graph of skeleton is constructed where two nodes are connected if there exist an edge between their corresponding triangles. In this representation a branching path could represent a limb. By carefully selecting pixels of triangles along a branch a body part is obtained. Since the skeleton-based method is very crude the paper proposes to use concept of blobs to better segmentation. Each body part is modelled using GMM. However, in some cases the human body might become occluded in such case the model driven method is used. In order to choose the most viable option from the model space, posture descriptor: "centroid context" is used. Then clustering method can be used to build the model space. Occasionally some postures have similar contour, which makes selection of best reference model problematic. The paper uses distance transformation or similarity or consistency of two pictures in sequence to match the best model pattern. Even though the proposed solution shows remarkably high accuracy it has some limitations. The camera which is used for capturing must

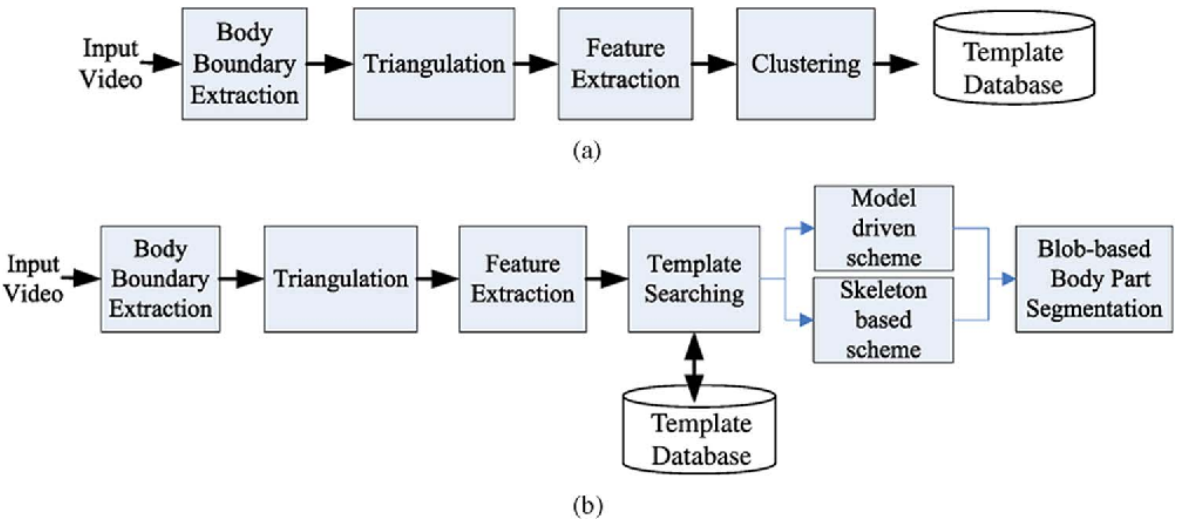

Figure 2.1: The proposed model in Segmentation of Human Body Parts Using Deformable Triangulation consisting of (a) training stage and (b) segmentation stage.

not be moved to successfully extract the human silhouette. Since background subtraction is used the background must be static all the time. Moreover, a dataset of key postures must be prepared beforehand in order to guide the model driven approach in case occlusion was detected. And lastly the pose segmentation behaves oddly when the top most part of human body is not head [7].

Another work was conducted by Deboeverie et al. in field of Physiotherapy. In this work they propose solution for reconstructing human body skeleton and part segmentation from 2d images. Grayscale images are segmented by grouping image pixels into regions, where each region is represented by geometric primitives, which represent parts of human body. This approach is achieved by using adaptive region growing algorithm. Region growing is the process of determining whether a new neighbouring pixel should be added to the region. Polynomial fitting is used to find a geometric primitive representing region of pixels. How well the primitive corresponds to region is denoted by fitting cost. Subset of pixels in region is used to compute the local fitting cost and set of all pixels in region is used to compute the global fitting cost. Local cost is used in process of region growing while global cost is used for adapting the shape of function. The process of region growing ends when polynomial degree of function reaches two. Part classification is based on shape of function representing a region. The shape is determined by signs of eigenvalues of the Hessian matrix and can be either convex, concave or saddle like. Human body skeleton can be easily estimated by finding the axes of minimum curvature. This approach can segment human pose only into three regions based on body part shape moreover can be easily distracted by shapes in background. The extracted skeleton is also very rough and contains many redundant parts which must be manually removed [8].

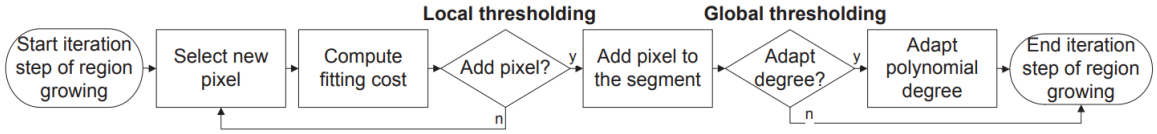

Figure 2.2: Flow chart depicting one iteration of adaptive region growing.

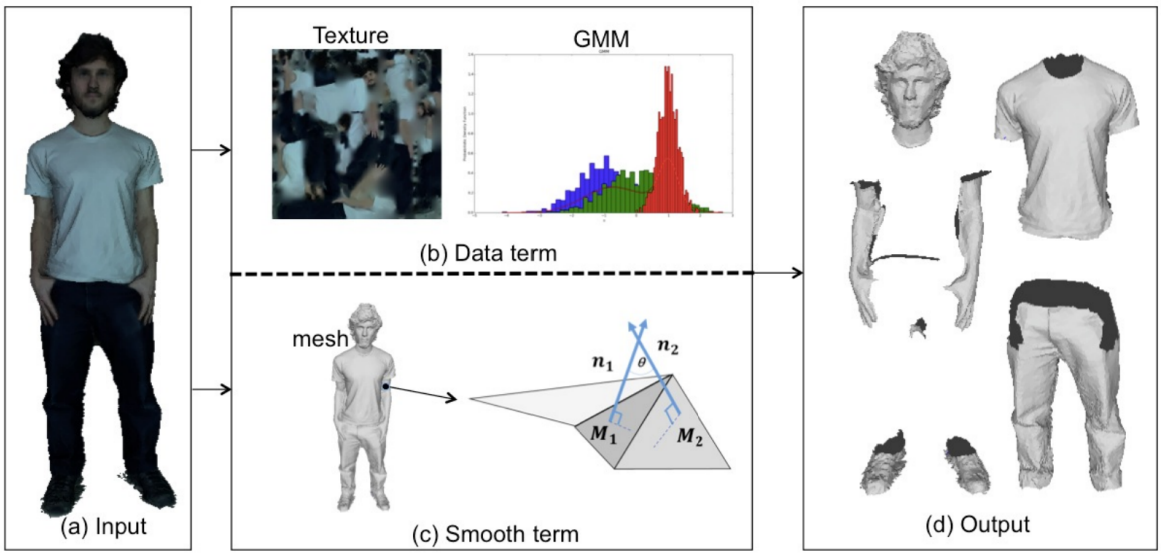

Figure 2.3: The proposed approach by Wang et al.. Given a human body mesh each vertex is labelled as one of five human body parts. Data term and smooth term model geometry and colour constraints in energy optimisation.

Paper published by Wang at al. is one of the few analytical segmentation methods aimed at 3D scanned human body. As a 3D scan a 3D mesh with texture is meant. In segmentation each triangle is labelled into one of five body parts. Their approach is based on energy optimisation consisting of two terms: data term and smooth term. Data term is based on probability that given face of mesh belongs to a specific class. The probability is based on assumption that in the colour space each collection of triangles follows a Gaussian Mixture Model distribution. The smooth term is based on the dihedral angle between two triangles of different class. Using prior information and built graph a GMM model is trained until the results are stable. If the prior information is poor, so are the results. The algorithm also has problem with converging if the mesh is oversimplified. Even though the results are relatively nice, they are heavily based on models' texture and geometry of clothes [24].

2.2 Machine Learning Approaches for Body Parsing

One noteworthy research which tries to bridge the gap between synthetic and real data was conducted by Lin et al. In their work they address the problem of manual labour

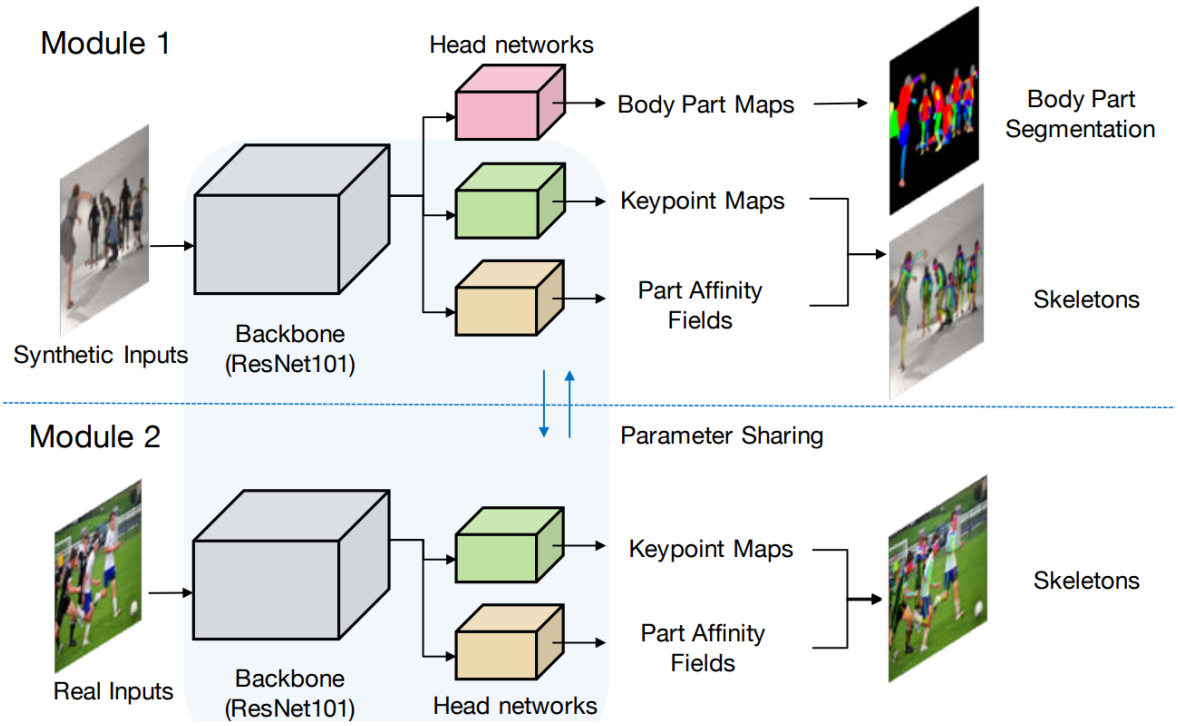

Figure 2.4: The model proposed by Lin et al in Cross-Domain Complementary Learning Using Pose for Multi-Person Part Segmentation.

required for pixel perfect ground truth annotations and limits of synthetic data generation. Their work introduced framework called cross-domain complementary learning with pose for multi-person which is able to segment both synthetic and real data. The architecture is relatively simple. It consists of two parts backbone and head networks. Backbone is ResNet101 consisting of five residual block. The output of backbone network is denoted as F = f(I), where I is an input image. Each head network consist of eight convolutional layers and can be denoted as B = HB(F), K = HK(F), P = HP(F) where B, K, P are body part segmentation maps, confidence keypoint maps and Part Affinity Fields, respectively. The architecture is trained on both synthetic data and real data using Adam optimiser. Synthetic data contain 3D pose labels and part segmentation labels while real data consist of only the 3D pose label. Using this approach the model learns pose estimation as auxiliary task on both synthetic and real world data in order to better generalise function for part segmentation on both types of input images [15].

Luo et al. contributed in field of adversarial learning for Human Parsing. This paper addresses the problem of local and semantic inconsistencies caused by pixel-wise classification loss. One possible solution would be to use adversarial training however training adversarial network on high-resolution images causes poor convergence. Macro-Micro Adversarial Nets were designed with limitations of adversarial networks

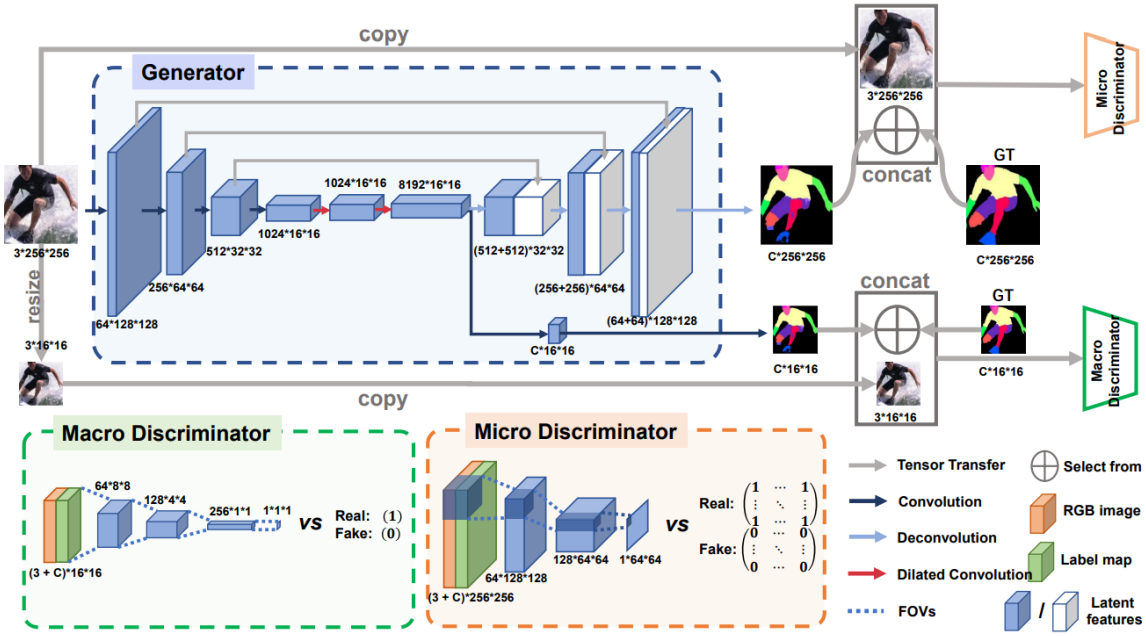

Figure 2.5: The proposed architecture capturing generator and adversarial networks proposed by Luo et al..

in mind. The network consist of generator utilising DeepLab-ASPP, Macro Discriminator and Micro Discriminator. Macro Discriminator is trained using low-resolution label map in order to improve high-level human characteristics produced by generator. Micro Discriminator is trained via high-resolution image to preserve local consistency in label maps [16].

Xia et al. proposed framework which leverages both segmentation and pose information to improve overall accuracy. The input image is firstly used to estimate segmentation and pose. Both pose and semantic part score are fed into Fully-Connected CRF to receive improved pose estimation. This in turn is fed into lighter Part FCN to gain improved estimation of semantic part segmentation [25].

2.3 Machine Learning Methods for Point Cloud Segmentation

Each neural network mentioned in this chapter was aimed for object classification, part segmentation of object or semantic segmentation of scenes [22, 21, 12]. Even though they were not proposed to solve the problem raised by human body segmentation we closely inspect their potential.

PointNet was a pioneering machine learning architecture, which unlike its predecessors directly consumed point clouds. The most widely used method for point cloud processing was either voxelization or projection into 2D images. This proved to be

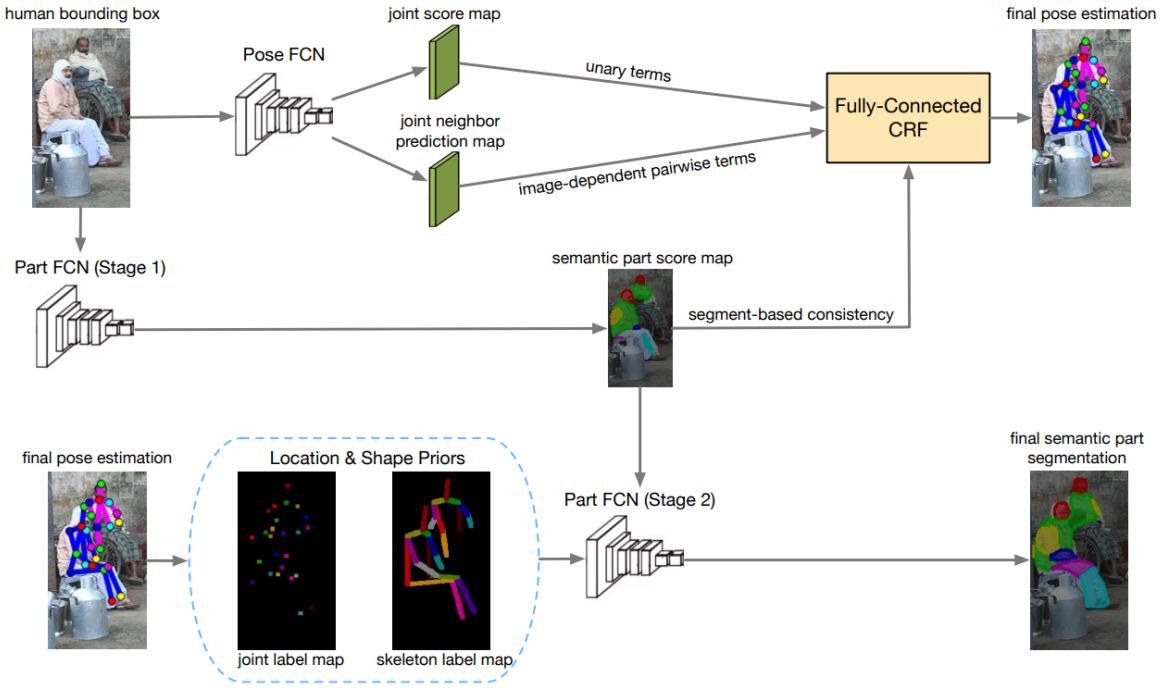

Figure 2.6: Adopted framework in Joint Multi-Person Pose Estimation and Semantic Part Segmentation. FCN stands for fully convolutional neural networks. CRF stands for conditional random field.

ineffective since 2D images both contained redundant information and partially lost spatial information. Voxelization produced data which lost some finer details and further learning required 3D convolutions with high computational cost especially for large scaled point clouds of scenes. For effective learning the architecture needed to take into consideration 3 main properties of point clouds: Point clouds are essentially unordered sets of points, which are feed into machine learning models as 2D array undesirably possessing implicit shape. Points are related by their distance and neighbouring points capture local structure. Point clouds can represent the same object under different transformations, therefore are invariant to transformation. Those 3 main properties are maintained using three key modules: max pooling layer, information combination structure and alignment networks [21].

PointNet++ is direct continuation of previously introduced neural model called PointNet. Even though PointNet captures global structure quite well it lacks the mean to capture local information induced by metric space. This hinders its ability in segmentation tasks. Convolutional architectures were the main inspiration which lead to processing of point clouds in hierarchical fashion. Standard data processed in CNN are defined in regular grids, images, where related pixels are closely grouped together. Such relation can be directly inferred from coordinates of pixels. On the other hand point clouds are unstructured and neighbouring points must be explicitly defined. In order to abstract local patterns point clouds are partitioned into overlapping regions

using Euclidean distance, then each region is fed into mini PointNet which produces local feature vector [22].

The aims of RandLa-Net is to effectively process large scale point clouds in tasks of both segmentation and classification. Most existing solutions for deep learning based segmentation and classification tasks of point clouds use algorithms which are computationally or memory inefficient. Despite many proposed architectures are effective for small sized point clouds comprised of thousands of points they fail to effectively process larger scale point clouds consisting of millions of points in real-time scenarios. The main culprits being either the complexity of sampling techniques used or inability of modules to capture wider context of points. The most efficient sampling method is random sampling, having computational complexity O(1). However random sampling can remove important features of sparse regions. RandLA-Net overcomes this problem by introducing local feature aggregation module [12].

Chapter 3

Proposed Solution

As was already stated in chapter 1 the process of segmentation of 2D images is essentially a classification of each pixel. Each pixel in image is assigned a value from finite set of labels. These labels split image into subsets sharing some common semantic meaning. The term segmentation can be applied more broadly to any medium consisting of finite number of elements for example unstructured point clouds. In our case we use 15 labels each designating a body part. By the process of segmentation we can split unstructured point clouds into regions depicting individual body parts.

3.1 Dataset

Synthetic datasets provide the comfort of easier ground truth generation however this comes at cost. Many proposed solution came to a problem caused by the so called domain gap [25]. The distribution of pixels in real world data is so different that the models trained using synthetic data have difficulty transferring knowledge from synthetic domain to real domain. For this very reason we opted for data captured in reality. Two datasets came to our attention: Berkeley Multimodal Human Action Database (MHAD) and CMU Panoptic Dataset. Both captured human body movement in various situations with ground truth skeletons for pose estimation. For this thesis CMU Panoptic Dataset was used since point clouds generated are of much higher resolution. With 3D animation skeletons available the generation of ground truth for segmentation is relatively simple. Furthermore for task of segmentation we only use point clouds representing single person, without background noise. We split the dataset into three complementary subsets for training, validation and testing.

3.2 Segmentation

In our experiments we use machine learning algorithms to approximate functions which segments point clouds into body part regions. Since each point must consider its position to its neighbours and each point is classified the segmentation architectures are generally more complex both in number of parameters and methods used in comparison to their classification counterpart. Moreover the proposed solutions must process raw unstructured point clouds without using conventional methods in 2D image segmentation. For the the approximation of our segmentation function we use three model architectures mentioned in chapter 2 and chapter 4 and compare the results on testing set.

The learning process takes place over multiple epoch. During each epoch we train our model using training subset. For each point cloud in training set the function predicts per point label. Using prediction and ground truth the training loss is computed. Based on loss the parameters of model are optimised to produce lower loss in further predictions. In order to assess whether our function generalises well to unseen data we use validation subset. If the model generalises well for unseen data both training and validation loss should be roughly the same. It should be noted that models parameters are never updated using validation loss.

Chapter 4

Implementation

In this chapter we mention technologies used for dataset pre-processing and framework used for building, training and testing selected neural architectures. Next we further elaborate selected machine learning algorithms and process of training.

4.1 Technologies Used

- **NumPy** is an open source python library used for scientific computing. The fundamental block of NumPy is ndarray which facilitate many mathematical, statistical and logical operations. Since python was not designed for numerical computing most of NumPy code is optimised and precompiled in C [6].
- **Pytorch** is an open source machine learning library for python. PyTorch defines multidimensional arrays called tensors which unlike NumPy arrays can be operated on by CUDA capable GPU. PyTorch library also contain modules for optimisation, building of computational graphs and backpropagation [4].
- **Kinoptic Dataset** is a shared dataset of point clouds dedicated for research purposes. The dataset consist of human body point clouds, captured by 10 synchronised Kinects installed in Panoptic Studio. The dataset also contains RGB videos and 3D skeletons. Currently there are footages of 54 sequences together 6 hours long [13].

4.2 Pre-processing of dataset

As stated in previous chapter for task of segmentation we use CMU Panoptic Dataset [13] concretely data from Range of Motion category. This category consist of data capturing individuals performing various simple poses and exercises. The individuals are captured by ten Kinect cameras. By using the proposed KinopticStudio Toolbox we

were able to extract point clouds by merging depth images from three cameras. Using available tools we partially removed floor and background noise, so only the human pose was captured. The point clouds were subsampled using farthest point sampling into samples of 2048 points. Since the Kinects were not perfectly synchronised some fast motions were incorrectly captured. Each pose has available 3D skeleton, consisting of 19 keypoints with spatial coordinates.

For annotation we used only the first 15 keypoints representing joints. To generate the ground truth we used a simple algorithm. For each point in pose we found the euclidean distance between each joint. From those 15 different distances we found the closest joint. This joint represents the specific category of point. Using this approach we were able to split the pose into 15 different regions, each representing segment of human body.

As each point is represented by three spatial coordinates of diverse values, this may create a bias in the process of learning if the values differ greatly. For this very reason we rescale our point clouds by min-max normalisation. Before feature scaling we center each point cloud by subtracting point c. Coordinates of this point are calculated per point cloud by averaging each coordinate of point in given point cloud. After centering our point cloud we scale each coordinate into the range $[-1, 1]$ by using the formula:

$$x' = a + \frac{(x - min\,(x))\,(b - a)}{max\,(x) - min\,(x)}$$

By substituting a and b with -1 and 1 respectively, we get:

$$x' = -1 + \frac{(x - \min\,(x))\ \ (1 - (-1))}{\max\,(x) - \min\,(x)} = \frac{2x - (\max\,(x) + \min\,(x))}{\max\,(x) - \min\,(x)}$$

Min and max value were found by finding the lowest and greatest values for each coordinate across the whole dataset. One other benefit of feature scaling is it can potentially improve gradient descent [23]. Each normalised point cloud and its corresponding annotation were individually saved as NumPy arrays. For the purpose of training 70% of dataset was randomly selected, for validation 20% were used, the remaining 10% were used for testing.

4.3 PointNet

The architecture of PointNet is generally made up of two T-Nets for coordinates transformation and feature transformation, plenty of shared multi-layer perceptrons and one max pooling layer. Three differing network architectures exist, each for different task: point cloud classification, segmentation and part segmentation. For the task of human body segmentation the later was used.

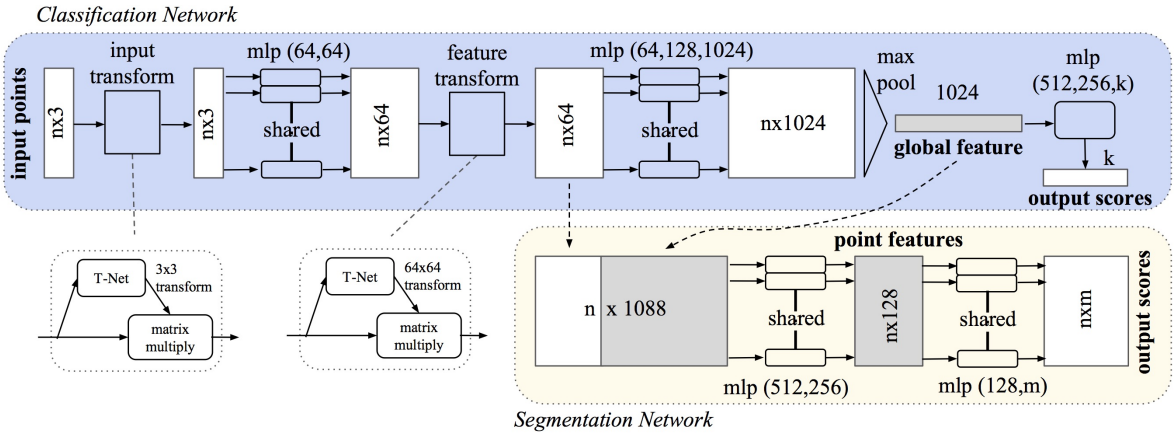

Figure 4.1: PointNet Architecture for both classification and segmentation, operating on n points. Prior feature map and global feature are concatenated for segmentation network.

Shared multilayer perceptrons

The architecture consists of multiple shared multilayer perceptrons, which are implemented using modules which apply one dimensional convolution which process feature matrix of shape nd_1, where n denotes number of points in point cloud, d_1 number of per point features and produces feature matrix of shape nd_2, where d_2 is size of feature vector of point. Each layer uses batch normalisation and ReLU as activation function.

Max pooling layer

To gain global signature of point cloud max pooling is used to process feature matrix of shape n $\times$ d into feature vector of size d. Using both shared multilayer perceptrons and max pooling the feature vector is invariant to point permutation, therefore point cloud produces the same global feature vector no matter the order of points.

Alignment network T-Net

T-Net are essentially small scaled point nets for classification lacking T-Nets. Each take point cloud of shape n $\times$ d and produce d $\times$ d transformation matrix which is multiplied with input point cloud to transform each point into canonical space.

Information combination

For object classification the global feature vector will suffice, however point cloud segmentation requires both global and local information. The local information is obtained as the intermediate feature matrix gained from previous shared multi-layer perceptrons. Both local and global features are concatenated to gain new point features aware of both local and global information. This approach however has its limitation, with

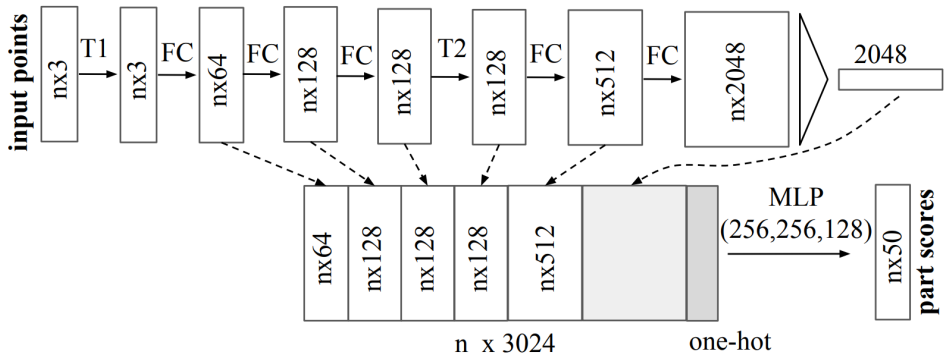

Figure 4.2: Point Net for part segmentation. Depicts how local features obtained from intermediate layers and max pooling layer are concatenated, to produce new feature tensor. FC is fully connected layer operating on each point. T1 and T2 are T-Nets. MLP is shared multilayer perceptron.

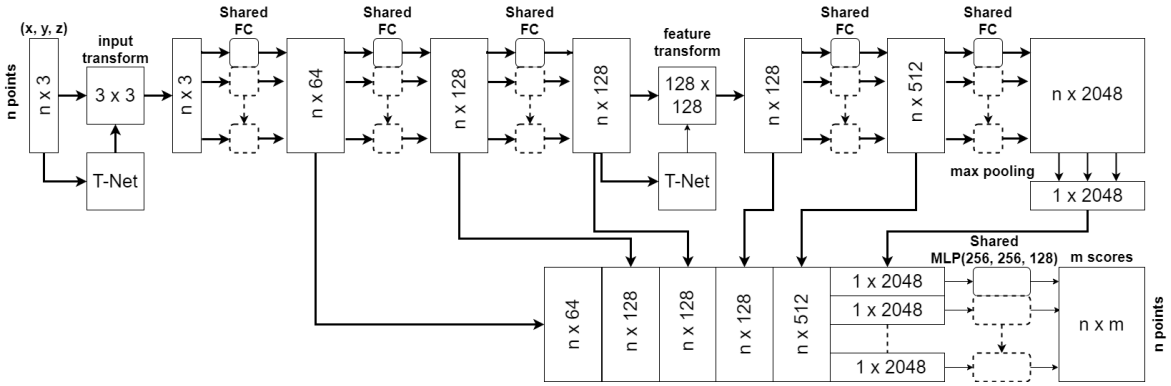

Figure 4.3: Our proposed modification of PointNet for part segmentation. The last two layers use dropout with keep ration 0.7.

many following works a different strategy for local feature aggregation were proposed [21].

4.4 PointNet++

PointNet++ architecture is made off stacks of set abstraction modules, feature propagation modules and skip link concatenation. The set abstraction modules consists of three layers: sampling layer, grouping layer and PointNet layer.

Sampling layer

Given a set of n points sampling layer chooses subset of points using iterative farthest point sampling. The first point is chosen at random then during each iteration new point is added which has the largest Euclidean distance from each point in previous subset until the subset of k point is found.

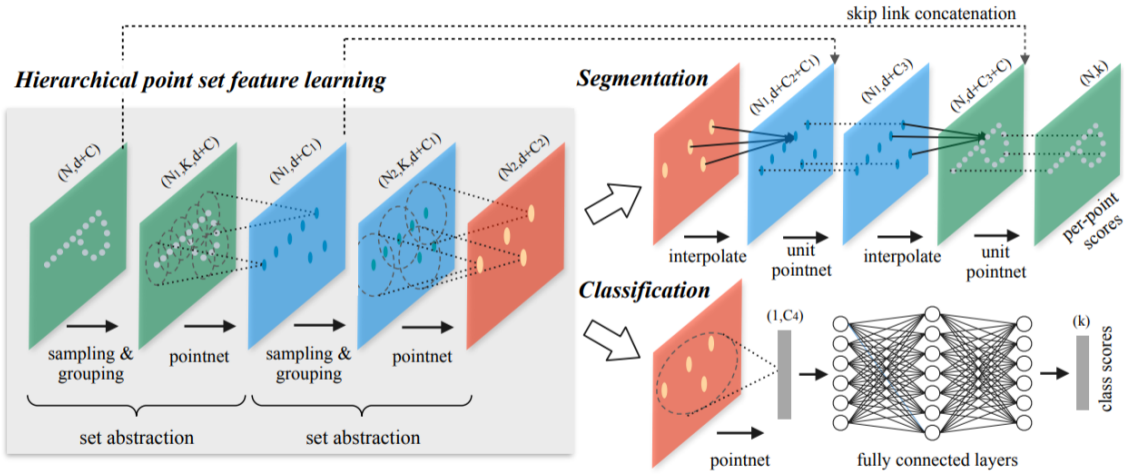

Figure 4.4: Illustration of key components of PointNet++ and their interactions. As an example points in 2D Euclidean space are used.

Grouping layer

This layers processes set of size n × (d + c), consisting of n points of d + C dimensions, where d denotes spatial coordinates and C number of additional features, and set of size m × (d) where m is number of centroids. The outputs is set of size m × k × (d + c), where k represents k nearest neighbours of each of m centroid. To find the k nearest neighbours ball query algorithm is used. Ball query finds neighbouring points which are within specified radius of centroid. If less than k points are found, the features of centroid are used to meet the limit. If more points exist they are simply dropped.

PointNet layer

The input for PointNet layer is tensor of shape m × k × (d + c), using mini PointNet across each of k neighbours, tensor of shape m × (d + e) is produced. Before being fed to network the coordinates of point are translated into relative position using respective centroid. The output of this layer is feature matrix capturing point-to-point relations.

Feature propagation

Set abstraction layers regularly subsample the original point cloud. In task of segmentation we want to obtain label for every single point. To receive point features of original size distance based interpolation is adopted to propagate subsampled features into original point cloud. The features are interpolated at coordinates of previous layer using inverse distance weighted average interpolation dependant on k nearest neighbours. Interpolated features are concatenated with features from skip link and fed into

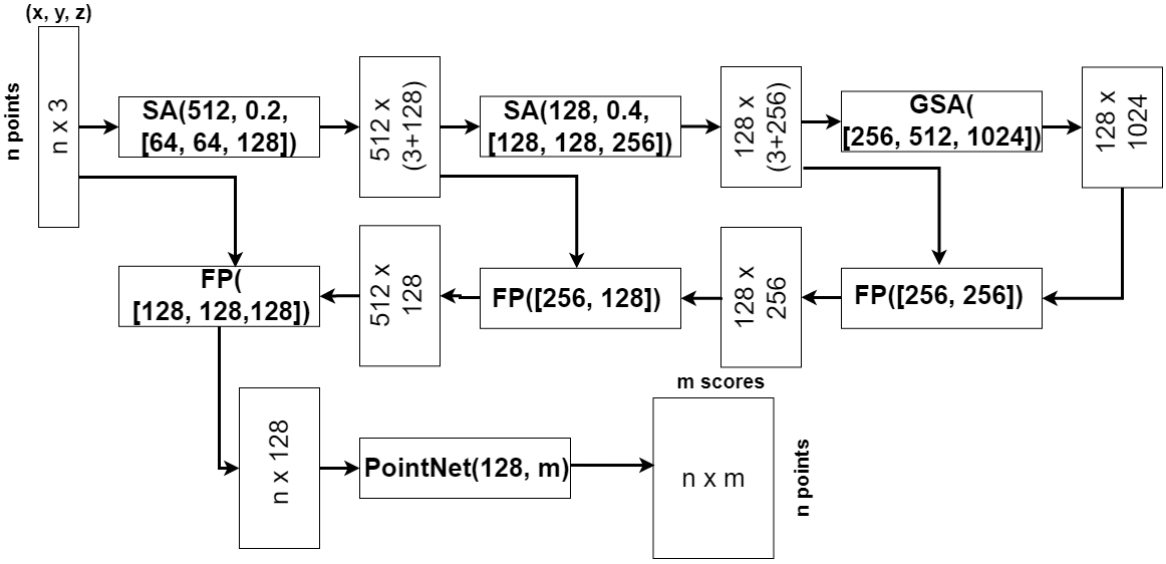

Figure 4.5: Our proposed modification of PointNet++. SA(k, r, [l1, ..., ld]) is set abstraction which samples input cloud using k centroids and ball radius of r. l1, ..., ld represent output dimensions of shared multilayer perceptrons of PointNet. GSA([l1, ..., ld]) is global set abstraction layer that process input tensor into vector via PointNet. FP([l1, ..., ld]) is feature propagation module which similarly uses PoinNet of specified dimensions. m is number of classes for segmentation

"unit pointnet". Inversed distance based interpolation on 3 nearest points using square distance for point x is calculated as:

$$f^{(j)}(x) = \frac{\sum_{i=1}^{3} w_i(x) f_i^{(j)}}{\sum_{i=1}^{3} w_i(x)};$$

$$w_i(x) = \frac{1}{d(x, x_i)^2}; j = 1, \ldots, C$$

Network for semantic and part segmentation

The architecture is depicted in figure 4.5. Each module recursively Applies PointNet on nested partitions of input set. During upsampling the architecture uses skip links to propagate features from previous layers.

4.5 RandLaNet

RandLA-Net consists of multiple feature aggregation modules stacked on top of each other. Each feature map is regularly downsampled by random sampling in encoding layer and upsampled in decoding layer using nearest-neighbour interpolation. Feature aggregation module consists of three blocks: local spatial encoding (LocSE), attentive pooling and dilated residual block.

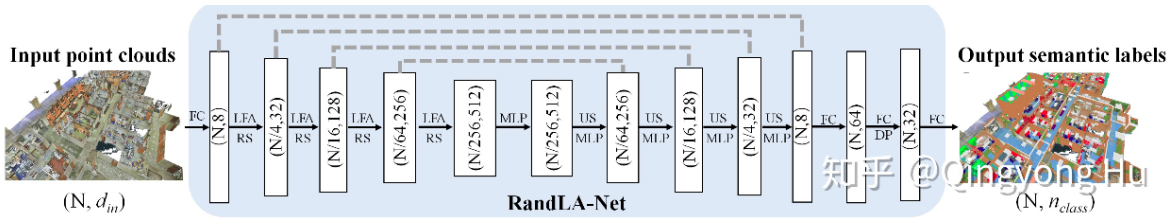

Figure 4.6: Architecture of RandLA-Net. FC: fully-connected layer, LFA: local feature aggregation module, RS: random sampling, MLP: shared multi-layer perceptron, US: upsampling.

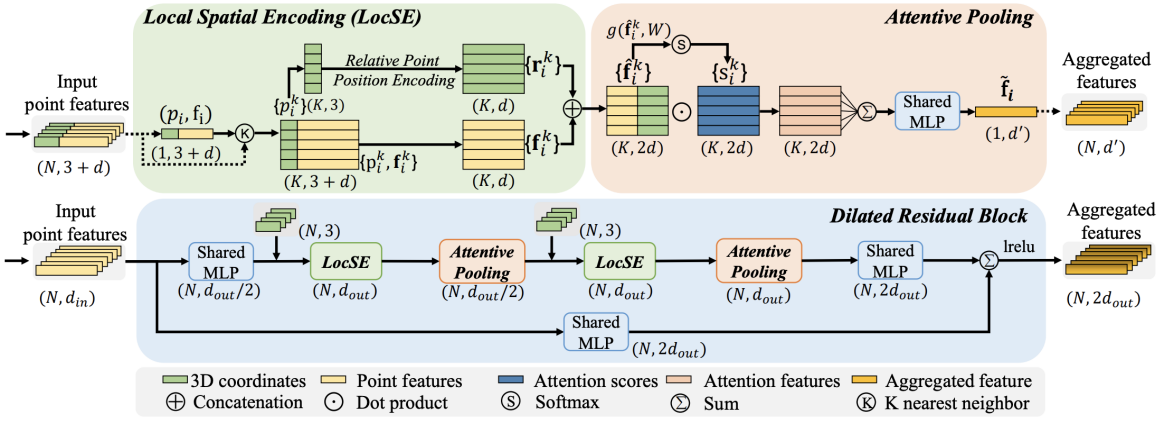

Figure 4.7: Key components of RandLA-Net. Local feature aggregation module at top. Dilated residual block at bottom.

Local Spatial Encoding

The goal of local spatial encoding unit is to embed spatial coordinates of neighbouring points so each point is aware of their relative position. For each point of point cloud K-nearest neighbours are found where each point has 3 spatial coordinates and d features. Nearest points are considered those which have the shortest Euclidian distance. Coordinates of point, coordinates of its i-th neighbour, relative coordinates and Euclidian distance are concatenated and processed by multilayer perceptron to align dimensionality of spatial features and original features. Both features are concatenated, gaining new feature tensor of shape k $\times$ 2d.

Attentive Pooling

The aim of this unit is to aggregate neighbouring features into single feature. Attention scores are computed for each feature gained from local spatial encoding block using shared MLP followed by softmax. Dot product of scores and their respective features is found and summed receiving feature vector of dimensions 1 $\times$ 2d. Shared MLP is used to produce feature vector of desired dimensionality.

Dilated Residual Block

In order to increase the receptive field for each point skip links are applied for multiple stacked feature aggregation modules. These skip links are used to retain important geometric features which might have been omitted by random sampling. Each input feature in dilated residual block passes through two different shared MLPs one feature map is used as skip link and the other passes through multiple feature aggregation modules and at the end trough final shared MLP. Those two feature tensors are summed and passes through leaky ReLU to produce output feature matrix.

4.6 Training

In our experiment we reimplemented PointNet and PointNet++ using publicly available code repositories as reference since most of the implementation details were vague, especially for PointNet++. Due to time constraints an already existing implementation of RanLA-Net was used. Each model was trained using the same dataset. To measure the performance of each network we used cross entropy loss. Cross entropy of probability distribution p_j and q_j over n classes is defined as [19]:

$$\mathbb{H}\left(p,\ q\right) = -\sum_{j=1}^{n} p_j \log\left(q_j\right)$$

Since p_j is mostly zero, except for some k, where $p_k = 1$ (because each object can belong only to single class, class k), the formula can be simplified as:

$$\mathbb{H}(p,\ q) = -p_k \log(q_k) = -\log(q_k)$$

Where q_k can be derived from softmax function, hence:

$$\mathbb{H}(p,\ q) = -\log\left(\frac{e^{q_k}}{\sum_{j=1}^{n} e^{q_j}}\right)$$

Therefore cross entropy loss is:

$$L = CE + R(w) = -\log\left(\frac{e^{s_p}}{\sum_{j}^{C} e^{s_j}}\right) + R(w)$$

Where R(w) is the regularisation loss. The regularisation term used in training of PointNet was computed as:

$$R(A) = 0.001 \cdot \|I - AA^T\|_F^2$$

Where A is the feature transformation matrix predicted by second T-Net. Both PointNet++ and RandLA-Net omitted regularisation loss. Each network was optimised using Adam optimiser with learning rate 10^{-3} and decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate was divided by 2 each 20 epochs as proposed in [21].

4.6.1 Data Augmentation

The data in our dataset consist of mostly the same poses performed by different actors. Since the point clouds are captured from the same angle and depict mostly perfectly captured poses without missing any limbs we propose a method to increase the variability of training samples. We implement various augmentation methods having analogues in 2D image augmentation, such as cropping a rotation. The process of data augmentation takes place before feature scaling and centering of point cloud. Since the same pose can be performed from various angles we simply rotate each point cloud along y-axis by random amount. Each point is rotated along the vertical axis using rotation matrix.

$$\begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} = \begin{bmatrix} x' & y' & z' \end{bmatrix}$$

Where α is the angle of rotation chosen from interval $[0,\ 2\pi]$ radians. We disregarded other axes since such transformation would be unnatural. To generate imperfections in point cloud and to motivate networks to learn spatial features of each limb we randomly cropped each point cloud. The algorithm consists of multiple steps. Since

point cloud cropping removes points in point cloud a minimum number of points must be specified, which we are willing to remove. For each axis we find the minimal and maximal coordinate across point cloud with regards to axis. Then we chose a random value t_i from interval $[-0.9d_i, 0.9d_i]$ where i is the selected axis and $d_i = |\min_i - \max_i|$. Each point is offsetted by t_i along axis i. If the coordinate i of point is greater than max_i or lower than min_i it is discarded. If such translation along an axis i would cause the point cloud to lose more points than acceptable the transformation is disregarded. The value 0.9 was used to prevent unreasonable translations, since translation by d_i would remove every point in point cloud. After point cloud is cropped each point cloud has potentially various number of points, which makes the process of machine learning impractical. Each point cloud must have the same number of points, since point clouds are processed in batches, which must keep constant size. To solve this problem each point cloud is subsampled using farthest point sampling so each point cloud has the same size. Since data augmentation takes place before normalisation and is random during each iteration of training it cannot be performed beforehand. This potentially makes the process of learning slower, however artificially increases the training dataset making the trained model more robust. Since the training set was augmented and different during each iteration we used only 60% of samples from dataset and split the rest evenly between validation and testing.

Chapter 5

Results

In this chapter we explain used metrics for evaluation of models, namely accuracy and mIoU, then we compare the results of best performing models of each network both qualitatively and quantitatively.

5.1 Used Metrics

In this section we elaborate our metrics for qualitative measurement of performance. We use mIoU since it is widely used metric for point cloud segmentation. Unlike accuracy mIoU penalises incorrect predictions in segmentation tasks.

Mean Intersection over Union (mIoU)

IoU is metric proposed by Jaccard [2, 1] for measuring similarity of two attributes associated with finite set of objects. Nowadays IoU is commonly used in segmentation tasks as evaluation metric [3, 7]. The general formula of IoU also known as Jaccard index is:

$$\sigma_{ik} = \frac{N\left(A_i \cap A_k\right)}{N\left(A_i \cup A_k\right)}$$

Where $N\left(A_i \cap A_k\right)$ is the number of times an object possesses both attributes i and k, while $N\left(A_i \cup A_k\right)$ is the number of times an object possesses one or both of attributes i and k. In segmentation tasks Jaccard index has the form:

$$IoU_j = \frac{|A_j \cap B_j|}{|A_j \cup B_j|}$$

Where IoU_j is the intersection over union of class j, with prediction A_j and ground truth B_j. Then mIoU of n classes can be calculated as:

$$mIoU = \frac{\sum_{i=1}^{n} IoU_i}{n}$$

Accuracy

To measure how accurately a model predicted an attribute j we used a simple metric. The accuracy of predicted attribute j is computed as:

$$acc_j = \frac{A_j \cap B_j}{|B_j|}$$

Where A_j denotes number of times it was predicted that object A possessed attribute j, B_j is the ground truth for object A. Then accuracy was computed ass summation of accuracies for each class:

$$acc = \frac{\sum_{i=1}^{n} A_i \cap B_i}{\sum_{i=1}^{n} |B_i|}$$

In our experiment accuracy was measured as the number of points correctly predicted out of all points in point cloud. Then overall accuracy was calculated as the mean of all accuracies during testing.

5.2 Evaluation

For evaluation each best performing trained model was tasked with segmenting of testing subset. The subset consisted of remaining 10% of samples from dataset which were not used during training and validation. During the second iteration of testing using augmented dataset, 20% of remaining samples were used, which were subsampled to 1024 points, centered and scaled. We compared both the accuracy, mIoU and number of parameters. Performance of each networks is captured in table 5.1. PointNet was trained for 73 epochs, with augmnetation for 100 epochs. PointNet++ was trained for 73 epochs and RandLA-Net for 65 epochs. The process of training was stopped when the improvements of trained networks were marginal. Overall the best performing architecture for human body segmentation was PointNet++ marginally outperforming PointNet with lower count of parameters. However even though PointNet++ had the best performance it has much higher computational complexity, potentially making it the slowest and least reliable for real-time segmentation. The retrained PointNet on augmented dataset had worse performance than PointNet trained without data augmentation. This might indicate slight overfitting of previous networks, since each was trained using 70% of samples from dataset. Despite the fact the retrained model had worse performance it might be more robust to imperfect data and transformations.

Table 5.1: Segmentation results on CMU Panoptic Dataset.

Name of Network	Number of Parameters (s)	accuracy	mIoU
PointNet	3366518	95.36%	90.32%
PointNet++	1388175	95.48%	90.5%
RandLA-Net	1300503	87.44%	76.86%
PointNet with data augmentation	3366518	92.42%	84.76%

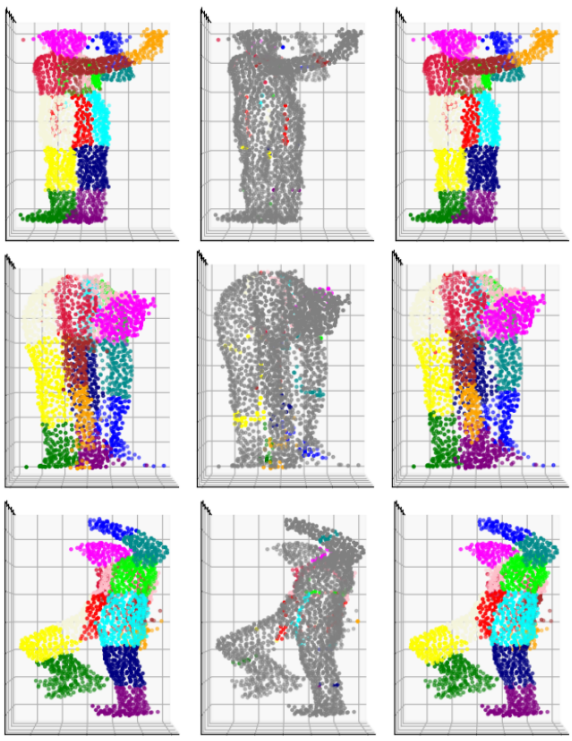

Figure 5.1: Human body segmentation achieved with PointNet. Left is prediction, right is ground truth, middle shows discrepancies

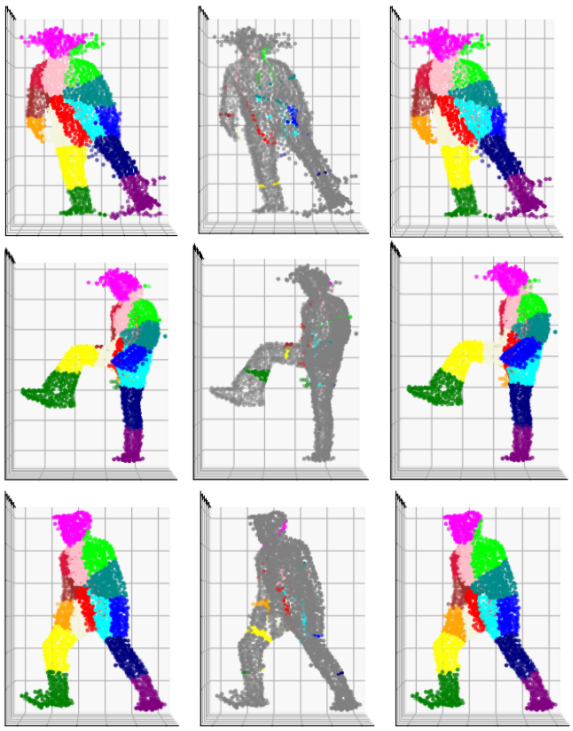

Figure 5.2: Human body segmentation achieved with PointNet++. Left is prediction, right is ground truth, middle shows discrepancies

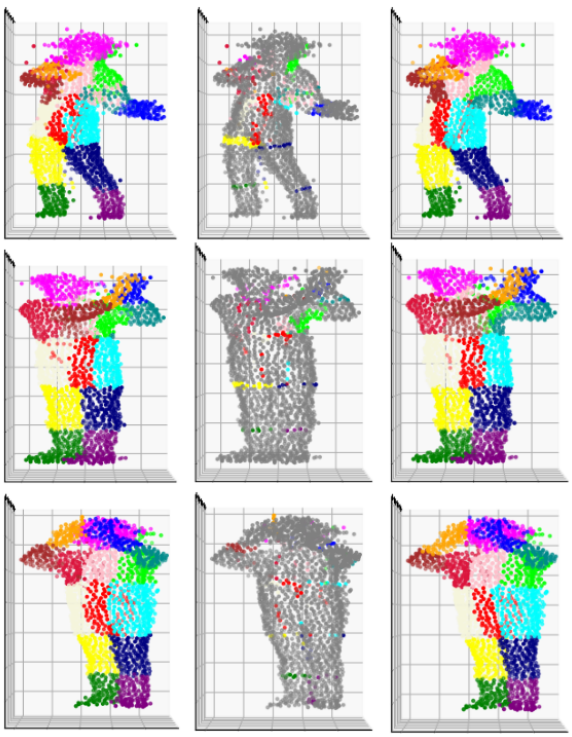

Figure 5.3: Human body segmentation achieved with RandLA-Net. Left is prediction, right is ground truth, middle shows discrepancies

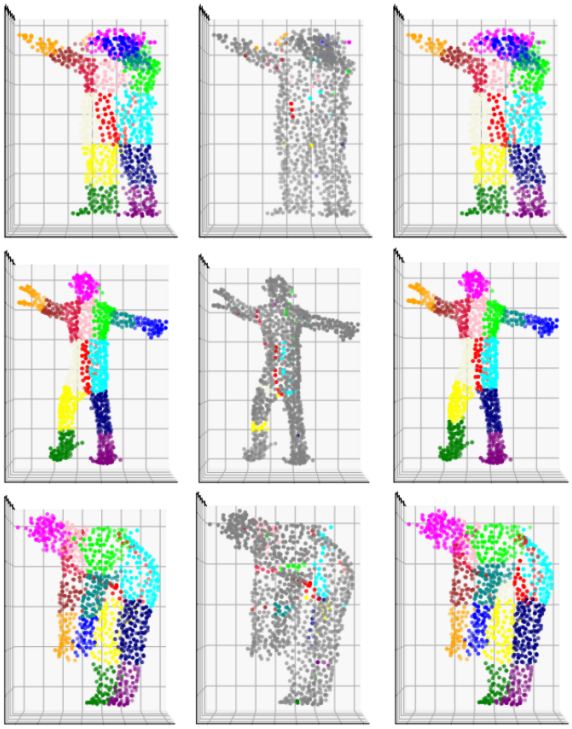

Figure 5.4: Human body segmentation achieved with PointNet trained on augmented data. Left is prediction, right is ground truth, middle shows discrepancies

Chapter 6

Conclusion

As of now, there are not many existing point cloud datasets aimed for human body segmentation and if so, they are comprised of synthetically generated point clouds. In this work, we used CMU panoptic dataset and used it for task of segmentation. The dataset is mostly aimed for pose estimation of different kind of activities, however we were able to use 3D animation skeletons to create ground truth for task of segmentation. Our method was relatively simple each joint was used to label the closest point. A better and non-trivial approach would be to use segments of 3D skeleton rather than joints. This way, the ground truth would bare information closer to human understanding of pose.

Furthermore, the proposed networks were trained only on scans of individual people and therefore are unsuitable if scan comprises of multiple persons. This problem could be solved by splitting it into two part: instance segmentation and part segmentation. Instance segmentation could split input point cloud into point clouds where each depicts single person which could be segmented into parts by our trained model, however this would be probably too inefficient and more general approach would perform the task better.

During our research we were able to train three neural networks. Since point clouds have some benefits over 2D images research is being conducted in automation of point cloud processing. Thanks to this motivation there are many proposed machine learning algorithm inspired by their image counterpart. Therefore there are new deep learning algorithms proposed each year, which outperform prior works and therefore might be more suitable for human body segmentation.

Human pose has a high degree of freedom and can express even more poses than dataset we used for training. Furthermore the dataset used depicted persons with relatively similar clothing and body shape. This could make the process of segmentation of our models difficult when the person segmented wore for example a dress. Additionally the proposed dataset consisted of human body scans lacking any background noise,

which again made our model unsuitable for real-life situations.

One possible method of preventing overfitting is data agumentation. We were unable to retrain each network using augmented data. The effect of point cloud data augmentation for human pose parsing should be explored more thoroughly.

Despite those negatives, we were able to show strong performance of different kinds of neural networks for human body segmentation, which is somewhat impressive since the aim of those networks was never human body segmentation. Additional research could be done in assessing how well neural networks perform in tasks of pose estimation from point clouds.

Further work could be done in proposing a benchmark for pose segmentation or perhaps a new dataset aimed for segmentation of human body in real-life scenarios.

Bibliography

[1] Course CS231n Convolutional Neural Networks for Visual Recognition convolutional-networks. `https://cs231n.github.io/convolutional-networks/`. Accessed: 2022-05-30.

[2] Course CS231n Convolutional Neural Networks for Visual Recognition neural-networks-1. `https://cs231n.github.io/neural-networks-1`. Accessed: 2022-05-30.

[3] Example of 2d convolution. `http://www.songho.ca/dsp/convolution/convolution2d_example.html`. Accessed: 2022-05-30.

[4] PyTorch documentation. `https://pytorch.org/docs/stable/index.html`. Accessed: 2022-05-30.

[5] What are point clouds? `tech27.com/resources/point-clouds/`. Accessed:2022-05-30.

[6] What is NumPy. `https://numpy.org/doc/stable/user/whatisnumpy.html`. Accessed: 2022-05-30.

[7] Chih-Chiang Chen, Jun-Wei Hsieh, Yung-Tai Hsu, and Chuan-Yu Huang. Segmentation of human body parts using deformable triangulation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40:596–610, 2010.

[8] Francis Deboeverie, Roeland De Geest, Tinne Tuytelaars, Peter Veelaert, and Wilfried Philips. Curvature-based human body parts segmentation in physiotherapy. In *VISAPP*, 2015.

[9] P. Návrat et al. *Umelá inteligencia*. Slovak University of Technology in Bratislava, 2002.

[10] Hao-Shu Fang, Guansong Lu, Xiaolin Fang, Xie Jianwen, Yu-Wing Tai, and Cewu Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. pages 70–78, 06 2018.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[12] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Agathoniki Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11105–11114, 2020.

[13] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[14] G. Stockman L. Shapiro. *Computer Vision*. Pearson, 2001. Accessed: 2022-05-30.

[15] Kevin Lin, Lijuan Wang, Kun Luo, Yinpeng Chen, Zicheng Liu, and Ming-Ting Sun. Cross-domain complementary learning using pose for multi-person part segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 31:1066–1078, 2021.

[16] Yawei Luo, Zhedong Zheng, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Macro-micro adversarial network for human parsing. In *ECCV*, 2018.

[17] Meysam Madadi et al. *Human segmentation, pose estimation and applications*. Universitat Autònoma de Barcelona, 2017.

[18] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[19] K. P. Murphy. *Machine Learning A Probabilistic Perspective*. MIT Press, 2012.

[20] Michael Nielsen. Neural networks and deep learning. `http://neuralnetworksanddeeplearning.com/`. Accessed: 2022-05-30.

[21] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[22] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[23] Xing Wan. Influence of feature scaling on convergence of gradient iterative algorithm. *Journal of Physics: Conference Series*, 1213(3):032021, jun 2019.

[24] Hanqing Wang, ChangYang Li, Zikai Gao, and Wei Liang. Joint labelling and segmentation for 3d scanned human body. *SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality: Modelling and Simulating Virtual Humans and Environments*, 2016.

[25] Fangting Xia, Peng Wang, Xianjie Chen, and Alan Loddon Yuille. Joint multi-person pose estimation and semantic part segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6080–6089, 2017.

Attachment

In electronic attachment attached to this thesis can be found the source code for PointNet, PointNet++, RandLA-Net, tools for data processing, visualisation and data augmnetation.

Due to copyright reasons the CMU Panoptic Dataset is not attached.