BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEEP NEURAL NETWORKS FOR LANDMARK DETECTION ON 3D MODELS

HLUBOKÉ NEURONOVÉ SÍTĚ PRO DETEKCI LANDMARKŮ NA 3D MODELU

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **TIBOR KUBÍK**
AUTOR PRÁCE

SUPERVISOR **Ing. MICHAL ŠPANĚL, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2021

Brno University of Technology
Faculty of Information Technology

Department of Computer Graphics and Multimedia (DCGM) Academic year 2020/2021

Bachelor's Thesis Specification

Student: **Kubík Tibor**
Programme: Information Technology
Title: **Deep Neural Networks for Landmark Detection on 3D Models**
Category: Image Processing

Assignment:

1. Get familiar with deep neural networks and their learning.
2. Get acquainted with current methods of detection of anatomical landmarks in 2D / 3D image data, or directly on 3D surface models. Focus on methods using neural networks.
3. Choose the appropriate method to detect the selected type of landmarks and prepare a data set for your experiments.
4. Implement the proposed method using existing tools for training deep neural networks.
5. Perform experiments on the data set and evaluate the results. Discuss possibilities of future work.
6. Create a short poster or video presenting your work, its goals and results.

Recommended literature:

- U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention, https://arxiv.org/pdf/1505.04597
- Evaluating deep learning uncertainty measures in cephalometric landmark localization. In: *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 2: BIOIMAGING*, https://www.scitepress.org/Link.aspx?doi=10.5220/0009375302130220

Requirements for the first semester:

- The first three items of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Španěl Michal, Ing., Ph.D.**
Head of Department: Černocký Jan, doc. Dr. Ing.
Beginning of work: November 1, 2020
Submission deadline: May 12, 2021
Approval date: October 30, 2020

Abstract

Landmark detection is a frequent step during medical data analysis. More and more often, these data are represented in the form of 3D models – an example is a 3D intraoral scan of dentition. Deep neural networks are an appropriate way of detecting landmarks in images. In terms of 3D data, the processing comes with high memory requirements and computational time, which does not meet the needs of medical applications. In this work, I propose a method that eliminates this problem and detects landmarks on the surface of polygonal models of jaws. Different architectures of neural networks, all of which are based on the U-Net architecture, are used in this work. The multi-view approach transforms the task into a 2D domain, where the suggested networks detect landmarks by heatmap regression from several viewpoints. Using a consensus method, final estimates from multiple views are predicted in 3D space. This work introduces experiments with two consensus methods – a centroid of predictions and a geometric approach based on the RANSAC algorithm and least-squares fit. Experiments have shown that a combination of Attention U-Net, 100 viewpoints, and RANSAC consensus method, is able to detect landmarks with an error of 1.20 ± 1.81 mm, while 94.01% of landmarks is predicted with an error of less than 2 mm.

Abstrakt

Detekcia významných bodov je častým krokom pri analýze medicínskych dát. Čoraz bežnejšie sú tieto dáta reprezentované vo forme 3D modelov, príkladom sú povrchové skeny zubného oblúka pacienta. Hlboké neurónové siete sú vhodný spôsob, ako detekovať významné body v obraze. V prípade 3D dát je však toto spracovanie časovo i pamäťovo náročné, čo nevyhovuje požiadavkám kladeným medicínskymi aplikáciami. V tejto práci navrhujem metódu, ktorá tento problém eliminuje a detekuje významné body na povchu polygonálnych modelov čeľustí. V metóde sú použité rôzne architektúry neurónových sietí, založené na architektúre U-Net. Viacpohľadový prístup presúva spracovanie do 2D, kde navrhnuté architektúry detekujú body regresiou tepelných máp z niekoľkých pohľadov. Pomocou konsenzus metódy je následne z týchto pohľadov určená konečná pozícia bodov v 3D priestore. V práci sú predstavené experimenty s dvoma konsenzus metódami – stredná hodnota predikcií a geometrický prístup založený na algoritme RANSAC a metóde najmenších štvorcov. Experimenty ukázali, že varianta kombinujúca Attention U-Net, 100 pohľadov a geometrickú konsenzus metódu je schopná detekovať významné body s chybou 1.20 ± 1.81 mm, pričom 94.01% predikcií dosahuje chybu menšiu ako 2 mm.

Keywords

landmark detection in 3D, polygonal models, multi-view neural networks, RANSAC, U-Net, heatmap regression

Kľúčové slová

detekcia landmarkov v 3D, polygonálne modely, viacpohľadové neurónové siete, RANSAC, U-Net, regresia tepelných máp

Reference

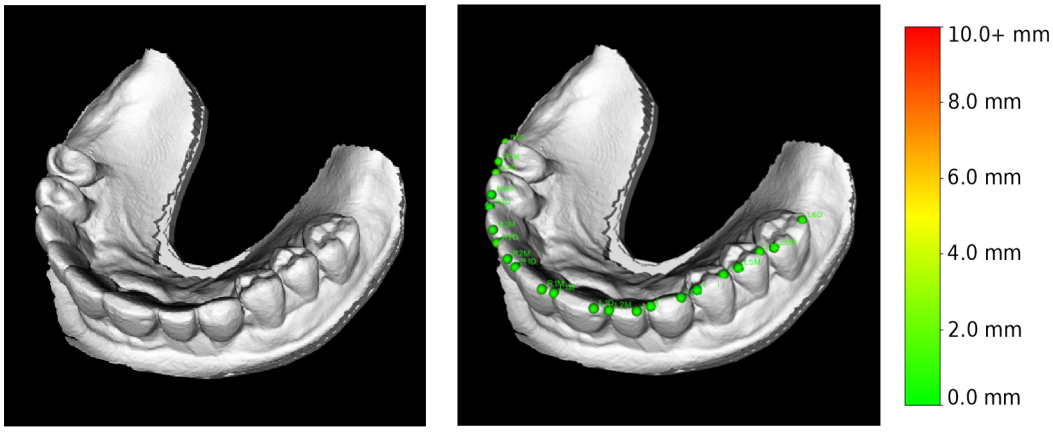
KUBÍK, Tibor. *Deep Neural Networks for Landmark Detection on 3D Models.* Brno, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Michal Španěl, Ph.D.

Rozšírený abstrakt

Úvod

Detekcia významných bodov je častým krokom pri analýze medicínskych dát. So stále zväčšujúcou sa dostupnosťou 3D zariadení sú medicínske dáta čoraz častejšie reprezentované vo forme 3D modelov. Táto reprezentácia je vhodná napríklad v digitálnej ortodoncii, keďže zubný oblúk pacienta naskenovaný vo forme povrchového modelu umožňuje jednoduchšiu tvorbu ukážky chrupu po dokončení liečby. Pri spracovaní 3D modelov neurónovými sieťami je však nutné počítať so zvýšením pamäťových nárokov a času potrebného na výpočet. Príkladom je operácia konvolúcie, pri ktorej sa časová zložitosť pre 3D vstup zvýši na $\mathcal{O}(n^6)$ zo zložitosti $\mathcal{O}(n^4)$ pri 2D konvolúcii.

Cieľom tejto práce je detekovať významné body na povrchu 3D modelov ľudských čeľustí, ako je možné vidieť na Obrázku 1. Ide o 32 bodov – dva pre každý zub na danej čeľusti. Motiváciou je zautomatizovať prácu ortodontistov, keďže v súčasnej dobe musia v procese tvorby modelu vyliečeného chrupu tieto body anotovať manuálne.



(a) Vstup metódy **(b)** Príslušný výstup

Obrázok 1: Ukážka vstupu a príslušného výstupu metódy. Metóda načíta polygonálny model ľudského chrupu, na ktorý automaticky nadetekuje významné body. Farba jednotlivých bodov v obrázku **(b)** popisuje radiálnu chybu, s akou metóda daný bod nadetekovala.

Popis riešenia

Keďže navrhovaná metóda je založená na neurónových sieťach, jej kľúčovým prvkom je dataset. V práci je použitý dataset ľudských čeľustí vo forme polygonálnych modelov. Tento dataset bol poskytnutý bez skutočných pozícií bodov ("ground truths"), preto súčasťou tejto práce je aj anotačný nástroj, ktorým je možné tieto hodnoty získať. Častým problémom medicínskych dát je ich nedostatok. Podobne je to aj v tomto prípade, keďže na natrénovanie, validáciu a vyhodnotenie systému je k dispozícii 269 povrchových modelov zubných oblúkov.

Navrhovaná metóda berie v úvahu oba aspekty, a to síce nedostatok dát a taktiež ich formu (3D polygonálne modely), ktorá značne zvyšuje pamäťovú a časovú náročnosť. Preto je zložená z troch primárnych častí:

- použitie viacpohľadových neurónových sietí [43],
- použitie architektúr neurónových sietí vychádzajúcich zo siete U-Net [35],
- nájdenie výsledného odhadu pozície z viacerých pohľadov vhodnou konsenzus metódou.

Viacpohľadový prístup je jedným zo spôsobov spracovania 3D modelov neurónovými sieťami. Model v scéne je vyhodnocovaný z niekoľkých pohľadov, vždy na 2D renderi scény. Metóda nepracuje s renderom geometrie, ale v každom pohľade je model renderovaný vo forme hĺbkovej mapy. Podľa použitého počtu pohľadov má metóda k dispozícii príslušný počet predikcií každého významného bodu vo forme obrazových súradníc.

V práci porovnávam dve konsenzus metódy, ktoré z niekoľkých predikcií vypočítajú jednu, finálnu pozíciu.

Prvá konsenzus metóda je založená na výpočte strednej hodnoty z daných predikcií. Tento konsenzus vyžaduje prvotnú konverziu obrazových súradníc do svetových súradníc, a to v každom z pohľadov. Pre dosiahnutie akceptovateľných presností bol stanovený predpoklad, ktorý očakáva predikcie zo všetkých pohľadov nasledujúce podobný vzor, čo znamená, že neurónová sieť nebude produkovať veľké množstvo nepresných predikcií. Inak by bola konečná predikcia ovplyvnená týmito chybnými predikciami.

Druhá konsenzus metóda je geometrická a využíva algoritmus RANSAC [11] v kombinácii s metódou najmenších štvorcov. Geometrická metóda v každom pohľade nevyžaduje konvertovanie predikcie do svetových koordinátov. Predikcia je interpretovaná ako polpriamka smerujúca z aktuálnej pozície kamery, pretínajúca pohľadovú rovinu v predikovanom bode. Táto skutočnosť výrazne urýchľuje detekciu významných bodov a zároveň prináša zvýšenú presnosť, keďže táto konsenzus metóda pri výpočte finálnej predikcie zanedbáva predikcie klasifikované ako extrémne prípady (tzv. “outliers”).

Ako bolo spomenuté vyššie, použité architektúry vychádzajú z architektúry U-Net. V práci sú použité tri modifikácie. Prvá z nich, BatchNorm U-Net, je podobná pôvodnej architektúre U-Net, no obsahuje navyše *batch* (dávkovú) *normalizáciu* medzi konvolučnou a ReLU vrstvou. Ďalšie použité architektúry sú Attention U-Net [28] a Nested U-Net [50].

Všetky architektúry sú trénované na regresnej úlohe, kde sú obrazové koordináty významných bodov reprezentované ako 2D Gaussovské rozloženia so stredom v danom bode. Ide teda o metódu regresie tepelných máp, ktorá sa pre tento typ úlohy ukázala ako najlepšia a používa sa často [10, 29, 30, 31, 49].

Experimenty

Súčasťou práce je sada experimentov, ktorá vyhodnocuje presnosť detekcie. Vrámci experimentov sú porovnané rôzne kombinácie navrhnutých architektúr a konsenzus metód. Taktiež sú vykonané experimenty s rôznym počtom pohľadov potrebných pre viacpohľadový prístup. Konkrétne ide o 9, 25 a 100 pohľadov, pričom bola stanovená hypotéza, že použitím väčšieho počtu pohľadov bude dosiahnutých lepších výsledkov.

Z experimentov vyplynulo, že z navrhnutých architektúr dosahuje najlepšie výsledky Attention U-Net. V každej z kombinácií konsezus metód a počtu pohľadov zaznemanala táto architektúra najmenšiu priemernú radiálnu chybu, ako aj mieru úspešnosti pri povolenej chybe 2 mm. Následne som analyzoval rozdiely pri použitých konsenzus metódach. Ukázalo sa, geometrický prístup dosahuje oveľa lepšie presnosti v porovnaní s priemerovaním predikcií. Spolu so skutočnosťou, že jeho vyhodnotenie spotrebuje menej výpočetného času som

prišiel k záveru, že táto konsenzus metóda je jednoznačne lepšia pre danú úlohu. Zvýšujúci sa počet pohľadov zvyšuje presnosť predikcií, avšak iba pri použití geometrickej konsenzus metódy, čím sa čiastočne potvrdila hypotéza. So zvyšujúcim sa počtom pohľadov sa generuje viac extrémnych prípadov, čo zhoršuje výsledky pri priemerovaní.

Najlepšiu celkovú presnosť teda dosahuje kombinácia Attention U-Net architektúry a geometrickej konsenzus metódy za použitia 100 pohľadov. Táto kombinácia dosahuje chyby 1.20 ± 1.81 mm, pričom miera úspešnosti predikcie významných bodov pri povolenej chybe 2 mm je 94.01%.

Súčasťou experimentov bolo taktiež pozorovanie, či je možné analýzou predikovaných tepelných máp určiť prítomnosť významného bodu na povrchu polygonálneho modelu. Hypotézou bolo, že tepelná mapa bodu neležiaceho na povrchu 3D modelu bude obsahovať veľmi nízku maximálnu hodnotu (blízku k nule). Naopak, ak je bod predikovaný s veľkou istotou, príslušná tepelná mapa bude obsahovať maximálnu hodnotu blízku k hodnote 1, keďže neurónové siete boli trénované regresiou tepelných máp s amplitúdou 1. Experimantami bolo zistené, že čisto analýzou tepelných máp je možné klasifikovať prítomnosť významných bodov s presnosťou 94.33%.

Zhrnutie výsledkov práce

V práci som navrhol spôsob, akým je možné detekovať významné body na povrchu 3D modelov, ktoré majú charakter medicínskych dát a pochádzajú z obmedzeného datasetu.

Za týmto účelom som vyhodnotil rôzne varianty architektúry U-Net, zistil som, aký má vplyv počet pohľadov pri viacpohľadovom prístupe na celkové výsledky a taktiež som ukázal chovanie dvoch konsenzus metód – priemerovania predikcií a geometrickej metódy založenej na algoritme RANSAC a metóde najmenších štvorcov. Navyše som vykonal analýzu maximálnych hodnôt tepelných máp, ktorej účelom bolo zistiť, či je táto informácia dostačujúca na tvrdenie, že daný významný bod sa na povrchu modelu naozaj nachádza alebo nie.

Ďalšie kroky tejto práce môžu smerovať k zníženiu chyby detekcie. V tomto ohľade by najviac pomohlo získať viac dát z menšinových tried. Ide konkrétne o povrchové modely chrupov, ktoré obsahujú tretie stoličky, keďže v trénovacej sade bolo takýchto modelov málo a tak je znížená celková presnosť metódy. Rovnako je možné vykonať dôkladnejšiu analýzu tepelných máp za účelom detekcie prítomnosti významného bodu na povrchu modelu. Táto analýza by mohla brať v úvahu počet predikcií, ktoré sú klasifikované ako extrémne, či fakt, že významné body sa na povrchu modelu vyskytujú vždy v dvojiciach.

Deep Neural Networks for Landmark Detection on 3D Models

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Michal Španěl, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Tibor Kubík
May 6, 2021

Acknowledgements

My deepest gratitude belongs to the thesis supervisor. The completion of this Bachelor's thesis would not have been possible without his suggestions and relentless guidance. Thank you, Mr. Španěl, for being my supervisor. Thank you for motivating me, for all your unwavering support, friendly attitude, and your time.

I would like to extend my gratitude to TESCAN 3DIM, s.r.o., which provided me with the dataset used to implement this work.

Contents

Chapter 1

Introduction

Localization of landmarks plays a crucial role in many tasks related to image analysis in medicine. Deep learning has demonstrated great success in this field, outperforming conventional machine learning methods. With the widespread availability of accurate 3D scanning devices, this task has moved into a 3D domain. This brings the possibility of automation of clinical application tasks that operate on 3D models, such as in digital orthodontics.

Taking into account the enormous increase in an input feature vector, a noticeable challenge has emerged. The time of computation of such deep neural networks is not suitable for a clinical application. 3D medical data analysis reckons with another challenge. The limited amount of medical data is a common struggle in medical image processing.

This work aims to develop solution for a fully automated orthodontics landmark detection on 3D jaw scans, which are represented as polygon meshes. The proposed method considers the limitation of the dataset and the need for low computational time, which is not standard in 3D deep learning. The method uses architecture designs that respect the dataset limitation, all of which are based on the U-Net architecture. Additionally, the task is trasfered into a 2D domain as it uses the multi-view CNN approach, where proposed networks detect landmark by heatmap regression from several viewpoints. Finally, the predictions from multiple views are used in a consensus method, where the final positions of landmarks in 3D space are detected. A comparison of two consensus methods is presented in this work – a method that calculates the mean value of multiple predictions and a geometric method based on the RANSAC algorithm and least-squares fit.

Conducted experiments have shown that the proposed method can detect orthodontics landmarks on surface models with an error of 1.20 ± 1.81 mm while 94.01% of detected landmarks achieve an error less than 2 mm. These results are obtained using the U-Net with integrated attention gates, 100 viewpoints, and RANSAC consensus method.

Firstly, this work introduces the base knowledge of dental anatomy and orthodontics in Chapter 2. The possible application of deep learning in digital orthodontics is then presented. Chapter 3 then explores current approaches to landmark detection. The U-Net architecture, its application in landmarking, and some significant modifications are discussed in this chapter as well. Lastly, an overview of CNN methods that process 3D data is depicted, focusing on the multi-view approach. The proposed solution for landmark detection is explored in detail in Chapter 4. This chapter outlines the overall method and closely describes its parts. The technologies used to implement this work are defined in Chapter 5. The custom annotation tool is introduced in this chapter too. Lastly, the evaluation of the proposed method can be found in Chapter 6.

Chapter 2

Landmark Detection in Orthodontics

To present the potential application of landmark detection in orthodontics, some basic terminology used in teeth morphology and orthodontics must be acquainted. This chapter firstly presents adequate vocabulary. Afterward, the malocclusion treatment process is described with a focus on automatic landmark detection applications.

2.1 Basic Dental Anatomy of a Human

The teeth in the human mouth form an arranged set called *dentition*. People have two dentition throughout life – the *primary* dentition during childhood and the *secondary* (also called permanent) dentition throughout adulthood. Teeth located on the upper jaw (also known as *maxilla*) form an arch called the *maxillary arch*. Analogically, teeth on the lower jaw (called the *mandible*) form an arch called the *mandibular arch*. Furthermore, each arch is divided into two halves resulting in the four-quadrant division of human dentition [33]. **For the rest of this work, the secondary dentition only will be taken into account.**

2.1.1 Tooth Identification Systems

Figure 2.1 illustrates the division and naming of teeth. Such naming is unnecessarily complex for routine clinic tasks and for storing in computer memory. To utilize a more sophisticated and practical way of tooth identification, unified notations were stated. Dentists use three main notation systems throughout the world.

The **Universal Numbering System** assigns numbers for teeth sequentially. Numbers from range 1 to 32 are used to distinguish the teeth, starting from the 3rd right molar on maxillary arch. The sequential pattern is followed until the 3rd left molar on maxillary arch, which is assigned by number 16. A similar pattern can be observed on the mandibular arch, starting from the 3rd left molar (number 17) until the 3rd right molar (number 32) [36].

Other notation systems are the World Dental Federation Notation and the Palmer Notation System. The notation of each of these systems is summarized in Table 2.1. **The Universal Numbering System is used throughout the rest of this work** to ensure consistency.

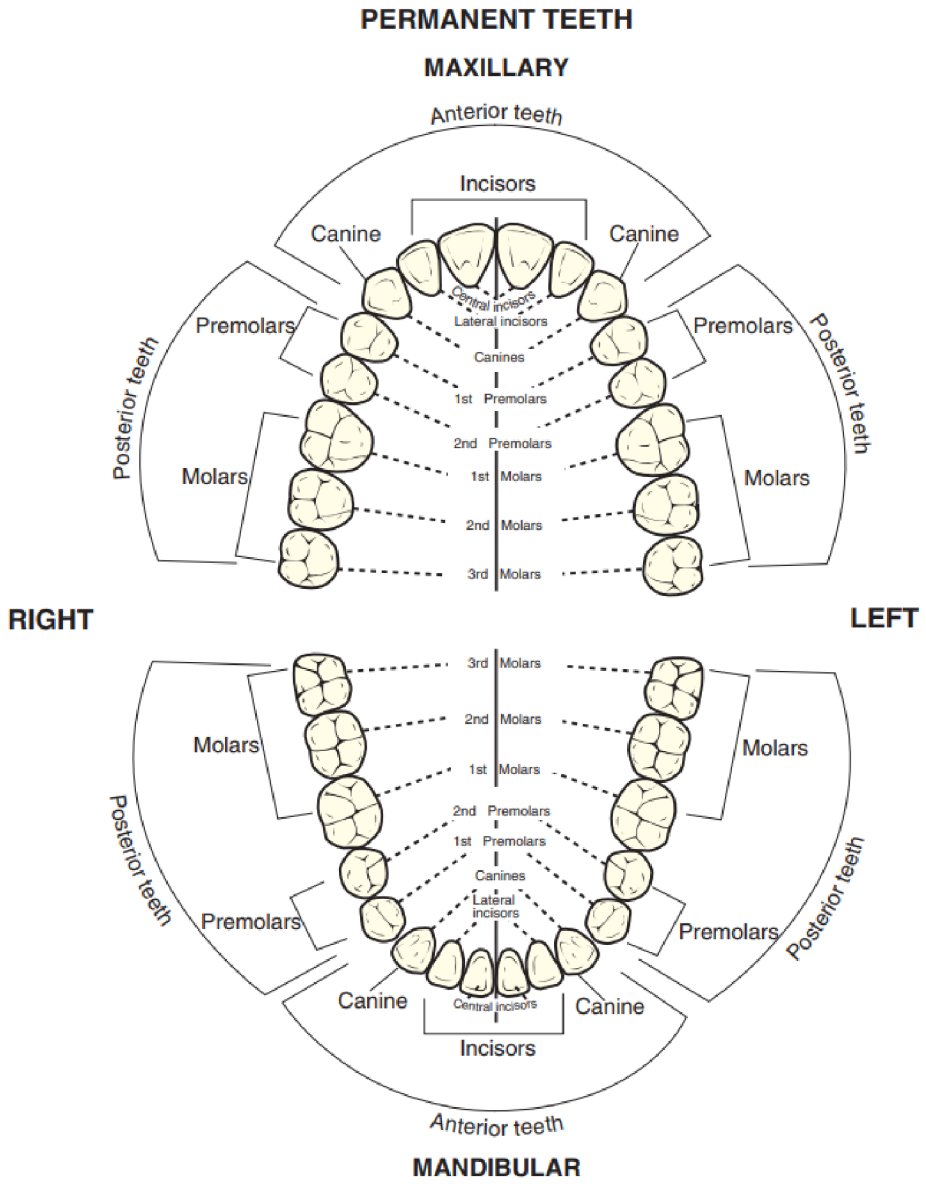


Figure 2.1: Scheme of maxillary and mandibular permanent dentition. It shows a complete dentition of a human adult. A complete secondary dentition contains 32 teeth, sub-divided into four quadrants. Furthermore, the quadrants are divided into different sub-groups. Adapted from [33].

2.1.2 Surfaces of the Teeth

Besides the tooth designation presented in Section 2.1.1, a reference to a more specific area of a tooth is often inevitable. For this purpose, there exist five types of surface per tooth:

- **Occlusal/incisal surface** is the chewing surface of a tooth,
- **Mesial surface** is the surface towards the midline of the dentition,
- **Distal surface** is the surface further from the midline,

- **Buccal/vestibular/facial surface** is the surface facing the cheek of the oral cavity,
- **Lingual/palatal surface** is the surface facing the inside of the oral cavity.

Tooth		**World Dental**		**Palmer**		**Universal**	
		Right	Left	Right	Left	Right	Left
Maxillary arch	Central incisor	11	21	1⌋	⌊1	8	9
	Lateral incisor	12	22	2⌋	⌊2	7	10
	Canine	13	23	3⌋	⌊3	6	11
	First premolar	14	24	4⌋	⌊4	5	12
	Second premolar	15	25	5⌋	⌊5	4	13
	First molar	16	26	6⌋	⌊6	3	14
	Second molar	17	27	7⌋	⌊7	2	15
	Third molar	18	28	8⌋	⌊8	1	16
Mandibular arch	Central incisor	41	31	1⌉	⌈1	25	24
	Lateral incisor	42	32	2⌉	⌈2	26	23
	Canine	43	33	3⌉	⌈3	27	22
	First premolar	44	34	4⌉	⌈4	28	21
	Second premolar	45	35	5⌉	⌈5	29	20
	First molar	46	36	6⌉	⌈6	30	19
	Second molar	47	37	7⌉	⌈7	31	18
	Third molar	48	38	8⌉	⌈8	32	17

Table 2.1: Summary table of most used tooth notations. The World Dental Federation Notation, the Palmer Notation System, and the Universal Numbering System are summarized.

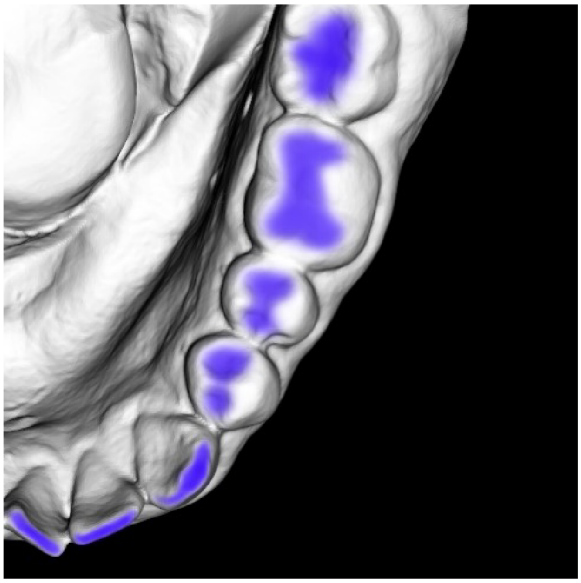


Figure 2.2: Example of occlusal and incisal surface (surfaces colored by purple).

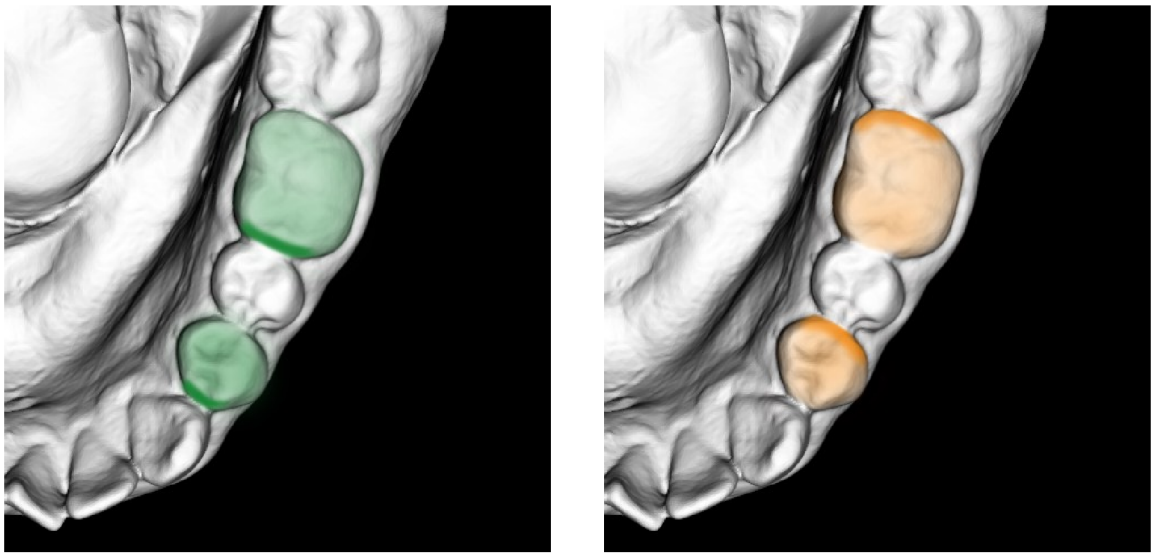


Figure 2.3: Left picture shows mesial surface – areas with high green intensity. Right picture shows distal surface – areas with high orange intensity.

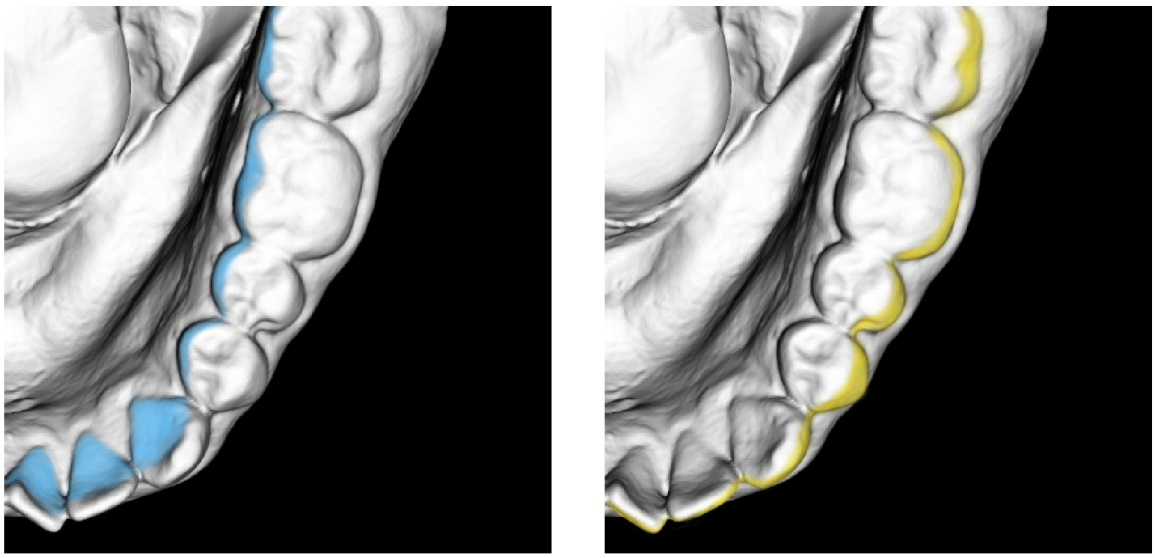


Figure 2.4: Example of tongue facing surfaces (left) and cheek facing surfaces (right).

In Figures 2.2, 2.3, and 2.4, detailed images for each surface group on the left mandibular quadrant are presented. The first mentioned figure shows two chewing surfaces – occlusal surface on teeth 18–20 and incisal surface on teeth 21–24. Teeth 19 and 21 in Figure 2.3 are used to demonstrate the mesial and distal surface.

2.2 Orthodontics

An ideal occlusion is defined as an anatomically perfect arrangement of the teeth [40]. In the wake of genetic predispositions and environmental factors, different types of anomalies are observed in human dentition. A branch of dentistry called orthodontics concerns the

diagnosis, interception, and treatment of such anomalies. An example of these anomalies is *malocclusion*. It is deviance from an ideal occlusion. As this deviance is common [34], there is a substantial desire for a modern and reliable treatment method.

2.2.1 Orthodontic Treatment

Malocclusion causes concerns related to health and quality of life, including speech quality, appearance, and psychological well-being. Malocclusion treatment (Figure 2.5) brings the benefits of aesthetics, speech, and dental health improvement.

The treatment process usually starts with the diagnosis during a routine dental examination. According to the type and severity, the malocclusion class is determined, and a proper treatment process is chosen. When a patient had been diagnosed with malocclusion, a model of his future ideal occlusion is created, which is how the treatment result is communicated [6].

Figure 2.5: Malocclusion treatment in process. The top image presents the dentition of a 12-year-old with a malocclusion. The bottom left image shows the treatment with the focus on maxillary arch alignment. The bottom right image presents the results of the malocclusion treatment on both dentition. Adapted from [40], edited.

2.2.2 Digital Orthodontics

Digitization has a significant impact on a variety of aspects of people's lives, including orthodontics. Digital orthodontics simplifies the malocclusion treatment and brings a new, faster communication with the patient [14].

With the rise of digital intraoral devices allowing 3D model capture, the orthodontic office workflow has simplified. As Taneva *et al.* present in their work [46]: *"In-office chairside or send to the lab, the digital models give the flexible options for designing and manufacturing a large range of dental restorations, implants, study models, and orthodontic*

appliances such as customized indirect brackets, archwires, expanders, aligners, retainers, etc."

As described in [32], the main advantages of digital models in orthodontics are reduced storage, effortless transfer and access, and simple integration into a patient's digital health record. Another remarkable advantage is the automation of routine tasks where machine learning comes forward. Machine learning systems provide good decision support that helps orthodontists work more efficiently with eliminated subjectivity and reduced variability [19].

2.3 Landmarks in Digital Orthodontics

Besides the aforementioned decision support, machine learning provides techniques for accurate landmark detection, more in Chapter 3. As shown in Figure 2.6, the digital workflow of an appliance creation is composed of four steps:

1. Patient's maxillary or mandibular arch is scanned into a 3D model.
2. **The appliance is designed. This step requires manual annotation of two landmarks on each tooth.** It is the most time-consuming step of digital workflow, thus offers tasks to be automatized.
3. 3D printing of designed appliance.
4. Delivery of appliance to patient.

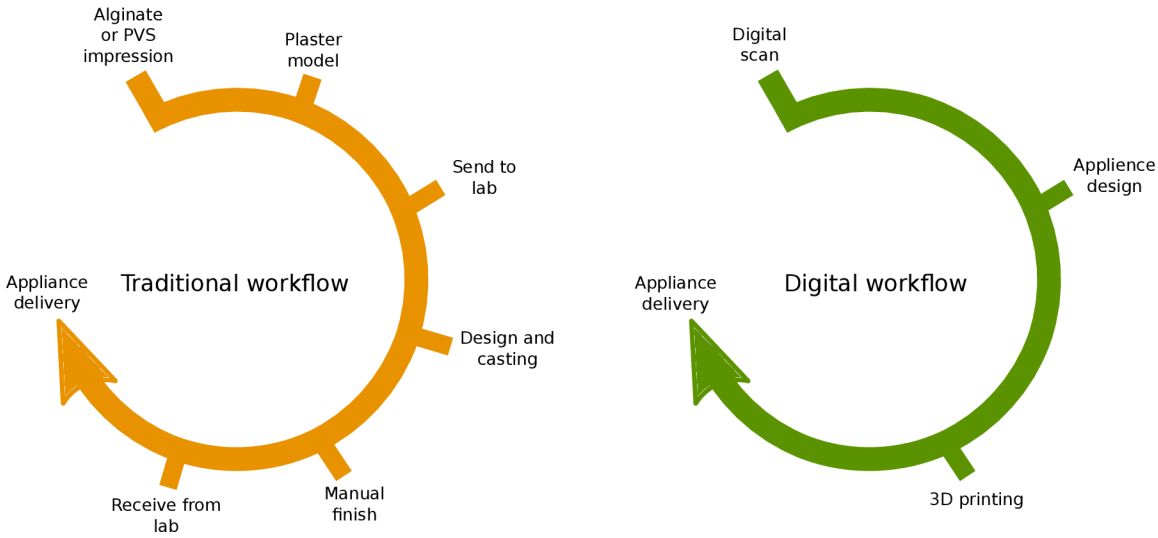


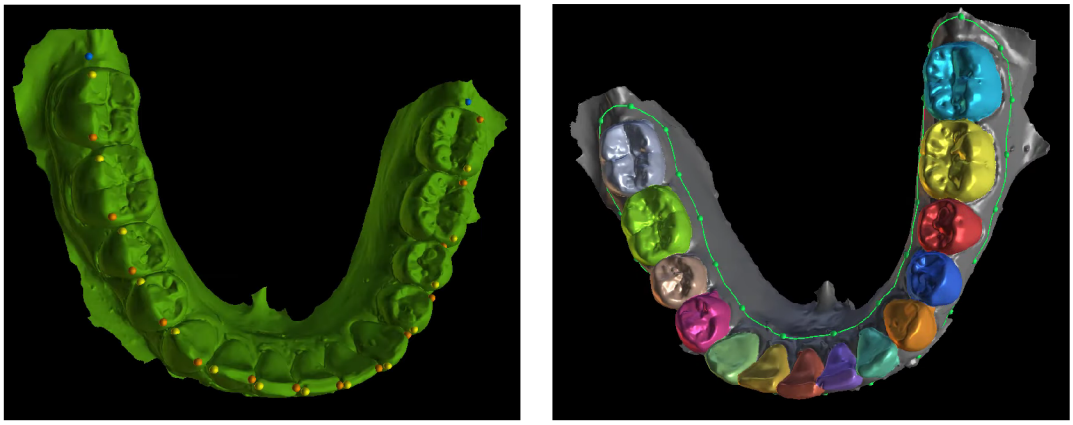
Figure 2.6: Traditional versus digital workflow in the orthodontic office. The digital workflow consists of a lower number of steps as the appliance design is modeled using an orthodontics software. The digital appliance saves a lot of time and space. Adapted from [46], edited.

Let's elaborate on the design process. In the existing orthodontics planning software, each tooth must be first annotated with two landmarks to form a model of ideal occlusion. **These landmarks define the mesial and distal location of each tooth and are placed on the occlusal surface on molars and premolars and the incisal surface**

on canines and incisors, as close to the cheek-facing surfaces as possible. In other words, **32 landmarks** must be placed on one arch in case of full dentition. Figure 2.7 shows an example of a professionally annotated model of the mandibular arch.

This annotation must be done precisely as its quality reflects in further automatic teeth segmentation. It is a time-consuming task, which must be done manually by the orthodontist. This substep of the digital workflow could be automatized by the deep learning methods for accurate landmark detection.

After the annotation is done, individual teeth are segmented. Manual post-processing is usually necessary due to segmentation errors. Each tooth's position in the scene can be individually changed, so the model of ideal occlusion is created [37].



(a) Annotation by an orthodontist **(b)** Resulting teeth segmentation

Figure 2.7: Depiction of a professionally annotated model and its further processing. Picture **(a)** shows a 3D model annotated by an orthodontist during the design process, and the post-processed teeth segmentation is depicted in the picture **(b)**. Each tooth has two landmarks placed on the incisal and occlusal surfaces. They define the mesial and distal tooth location. This information is subsequently used in the aforementioned segmentation. Note that the resulting segmentation is here just for illustration of further landmark placement usage. It is not a point of interest of this work. Adapted from [37], edited.

Chapter 3

Overview of the Current State

Image landmark detection has been a fundamental step in many computer vision tasks for years. One of these tasks is the detection of anatomical landmarks, for example, the detection of cephalometric landmarks in the cephalometric analysis [21, 49] or the detection of landmarks in X-Rays of the pelvis [3] or cardiac CTA scans [27]. The usage of accurate automatic landmark prediction in the medical field is simply wide-ranging, which motivated people to solve these problems years before the widespread popularity of deep neural networks in 2015 [22].

Early studies in this area focused primarly on the classification of bounding boxes containing landmarks [18] or voxels' classification [8, 25]. These tasks typically exploit only the local image information and used conventional machine learning approaches.

Rather than focusing on a single voxel of interest, the classification and determination of the volume of interest were combined to reduce computational costs [23]. Furthermore, classification was not the only way of detecting landmarks. Regression was used to predict landmark points [12, 15] as well.

Hough forests were used for landmark detection as they combined regression and classification. It was shown [9] that this combination leads to better results compared to regression only.

As convolutional neural networks (CNNs) gained in popularity, more and more scientific papers concerning their usage in landmark detection emerged. Some of these methods detected the landmark position directly from its coordinates, regressing its x and y values. For example, Sun *et al.* [45] adopted cascaded convolutional neural networks for facial point detection. Lv *et al.* [24] proposed a regression in a two-stage manner, still locating landmarks directly.

This chapter describes proven methods that use convolutional networks to detect landmarks and keypoints. The most significant one is the regression of heatmaps. Afterward, the U-Net network is presented together with its usage in landmarking and possible modifications. Lastly, a summary of the approaches of 3D data processing by CNNs is presented, with the focus on the multi-view approach.

3.1 Heatmaps in Landmarking

Pfister *et al.* [31] worked on a model that regresses human joint positions. Instead of directly regressing the (x, y) joint position, they regressed a joint position's heatmap. During the

training, the ground truth labels are transformed into heatmaps by placing a Gaussian with fixed variance at each of the joint coordinates. This can be seen in Figure 3.1.

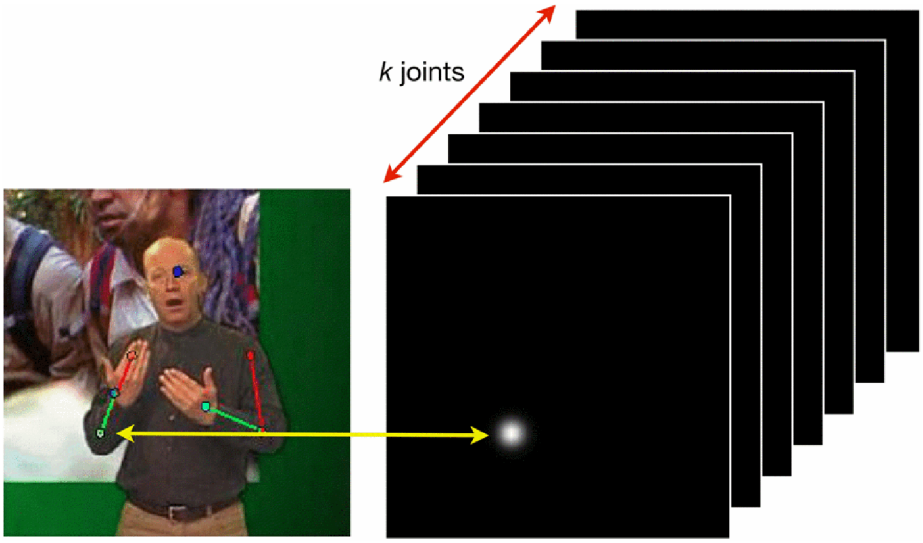


Figure 3.1: Landmarks representation by heatmaps. The network learns on a heatmap with a synthesized Gaussian with a fixed variance centered at the ground truth joint position for each of k joints. Adapted from [31].

They denoted the training example as a tuple $(X,\ y)$, where X is the input image and y stands for the coordinates of k joints located in image X. Furthermore, the training data were denoted as $N\ =\ \{X,\ y\}$ and the network regressor as ϕ. Then, the training objective is the estimation of the network weights λ:

$$\arg\min_{\lambda} \sum_{(X,y)\in N} \sum_{i,j,k} \|G_{i,j,k}(y_k) - \phi_{i,j,k}(X,\lambda)\|^2 \tag{3.1}$$

where $G_{i,j,k}(y_i) = \frac{1}{2\pi\sigma^2} e^{-[(y_k^1-i)^2+(y_k^2-j)^2]/2\sigma^2}$ is a Gaussian centered at joint y_k with fixed σ. Using this approach, the last convolutional layer's output is a heatmap represented as a fixed-size $i \times j \times k$-dimensional matrix.

On top of the appliance of spatial fusion layers and optical flow, they discussed the benefits of regressing a heatmap rather than $(x,\ y)$ coordinates directly. They concluded that the benefits are twofold: (i) the process of network training can be visualized in such a way that one can understand the network learning failures, and (ii) the network output can acquire confidence at multiple spatial locations. The incorrect ones are slowly suppressed later in the training process. In contrast, regressing the $(x,\ y)$ coordinates directly, the network would have a lower loss only if it predicts the coordinate correctly, even if it was "growing confidence" in the correct position.

This approach seemed alluring for people in the medical image processing community. Inspired by this method, Payer *et al.* [30] presented multiple architectures that detect keypoints in X-Ray images of hands and 3D hand MR scans. They affirmed that it is possible to achieve state-of-the-art localization performance in both 2D and 3D domains while dealing with medical data shortage by regressing heatmaps. Outline of thier method is depicted in Figure 3.2.

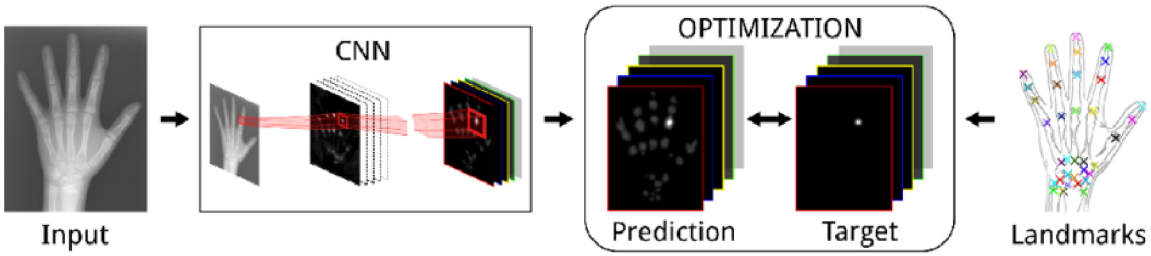


Figure 3.2: Overall framework for hand keypoints detection. In this work, the heatmap regression was used to predict landmarks on hands. In the optimization step, the local appearance of a landmark is combined with the spatial configuration of all other landmarks. This minimizes ambiguities caused by similarly-looking landmarks, e.g., fingerprints. Adapted from [30].

3.2 U-Net Network

U-Net network [35] was initially presented as an architecture for biomedical image segmentation. Yielding a u-shaped design, it can output a class label assigned to each pixel of the input image.

3.2.1 Network Architecture

The main idea of this architecture is its division into two parts, as shown in Figure 3.3: (i) the contracting path (left side), also called the encoder or the analysis path with the typical architecture of a convolutional network and (ii) the expansive path (right side), also called the decoder or the synthesis path. The contracting path consists of repeated blocks of two 3x3 unpadded convolutions, each followed by a rectified linear unit activation function (ReLU) [26] and a 2x2 max pooling for downsampling. Each downsampling step doubles the number of feature channels. On the contrary, the expansive path performs the upsampling and a 2x2 "up-convolution", that halves the number of feature channels. A correspondingly cropped feature map from the left part is concatenated at each level, followed by two 3x3 convolutions and a ReLU. Finally, a 1x1 convolution maps 64 feature channels into the desired number of classes. This original U-Net architecture with an almost symmetrical u-like style demonstrated excellent segmentation results on multiple segmentation tasks.

The original U-Net, however, is not the only architecture of its type. The immense growth of U-Net papers since 2017 has brought many modifications such as the Nested U-Net [50], Attention U-Net [28], or Residual U-Net [1]. The most important modifications, along with hundreds of examples of U-Net usages, are summarized in [39].

3.2.2 U-Net for Landmark Detection

In the wake of the success in segmentation tasks, the U-Net with different modifications found its usage in landmark detection. As presented in [30], this model design performed very well in combination with the heatmap regression approach. The main idea is to use the encoder-decoder approach to get a heatmap for each landmark of the exact resolution as the input image.

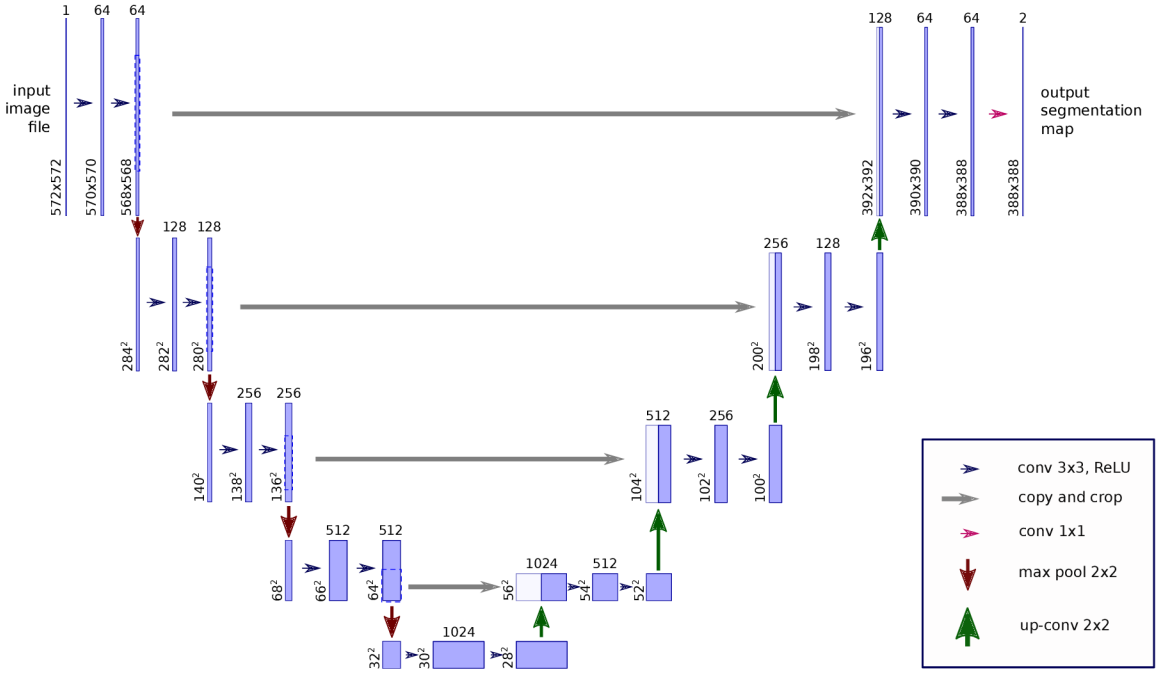


Figure 3.3: U-Net architecture. The U-Net is an encoder-decoder architecture initially presented for biomedical image segmentation. The number on top of the boxes indicates corresponding feature channels. The x, y size is provided at the bottom-left corner. An input image of size 572 × 572 is downsampled to 32 × 32 image. The decoder part enlarges the image to produce a segmentation map of size 388 × 388. Introduced by Ronneberger *et al.* in [35].

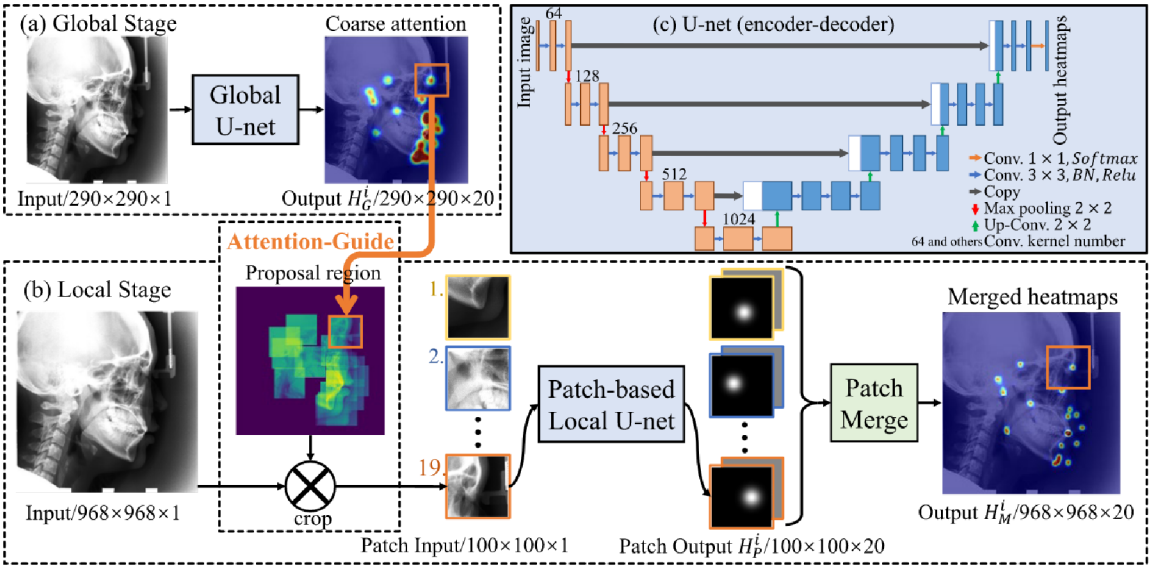


Figure 3.4: Overall structure of global and regional stage approach using U-Net. (a) **Global stage**. During the global stage, coarse attention is created and later used as a guidance in the local stage. (b) **Local stage with attention-guide**. This stage uses a patch-based U-Net model. Guided by the coarse attention from the Global stage, local stage searches in the proposal regions, regressing the heatmap patches in a high resolution. (c) **Modified U-Net structure for both global and local stages**. Adapted from [49].

To design a fully symmetric architecture, they used fixed linear upsampling kernels in the decoder part. Zhong *et al.* [49] used U-Net in a two-staged framework to detect 2D cephalometric landmarks. The first, "global stage" regresses the global heatmaps. Consequently, the information from the global stage is used in the patch-based local model. See Figure 3.4 for the illustration of the overall structure of their system.

3.2.3 Attention U-Net

One of the U-Net modifications is the Attention U-Net proposed in [28]. This architecture extends the original U-Net by the usage of *Attention gates* (AGs). The motivation for the AGs integration is an increase in accuracy. AGs progressively suppress feature responses in irrelevant regions without any additional supervision. In the U-Net model, they are incorporated to highlight salient features passed through the skip connections. Information from coarse-scale is then used in gating to disambiguate noisy responses in skip connections. Figure 3.5 shows the Attention U-Net model in a block diagram. It also depicts the schematic of the attention gate.

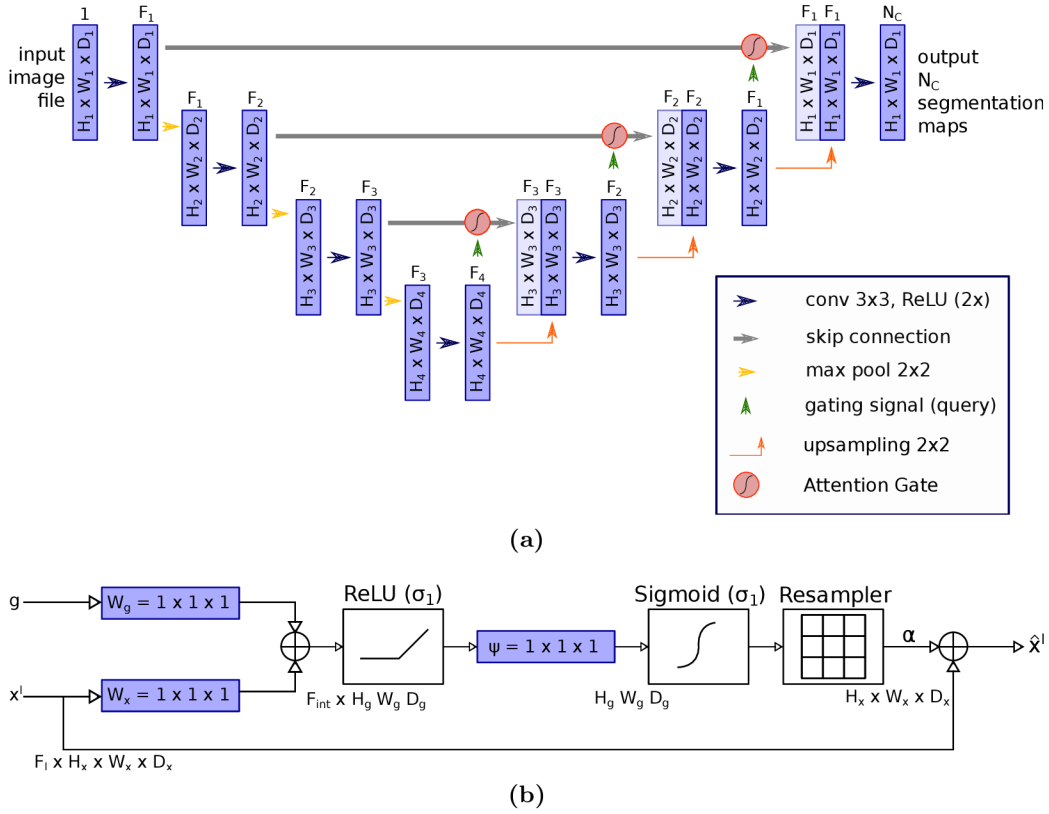


Figure 3.5: (a) A block diagram of the Attention U-Net model. The input image is downsampled by a factor of 2 on each level of the encoder part. Attention gates filter the skip connection features. **(b) Attention gate.** Input features x^l are scaled with attention coefficients α. α identify salient image regions and prune feature responses to preserve the relevant activations only. Gating signal g is collected from the coarser scale. The output of an AG is the product of input feature maps and attention coefficients. Originally presented in [28].

3.2.4 Nested U-Net

Zhou *et al.* [50] presented a U-Net modification called Nested U-Net, where they redesigned the skip pathways. The encoder and decoder parts of the network are connected through a series of nested, dense skip pathways. The aim is to reduce the semantic gap between the feature maps of the encoder and the sub-networks. Authors argue that to capture the fine-grained details of the foreground objects more effectively, the gradual enrichment of high-resolution feature maps from the encoder prior to fusion with the corresponding semantically rich feature maps from the decoder part should be ensured. Additionally, they state that the optimizer would deal with an easier learning task, as the concatenated feature maps are semantically similar. Figure 3.6 shows the architecture design and how it extends the original U-Net.

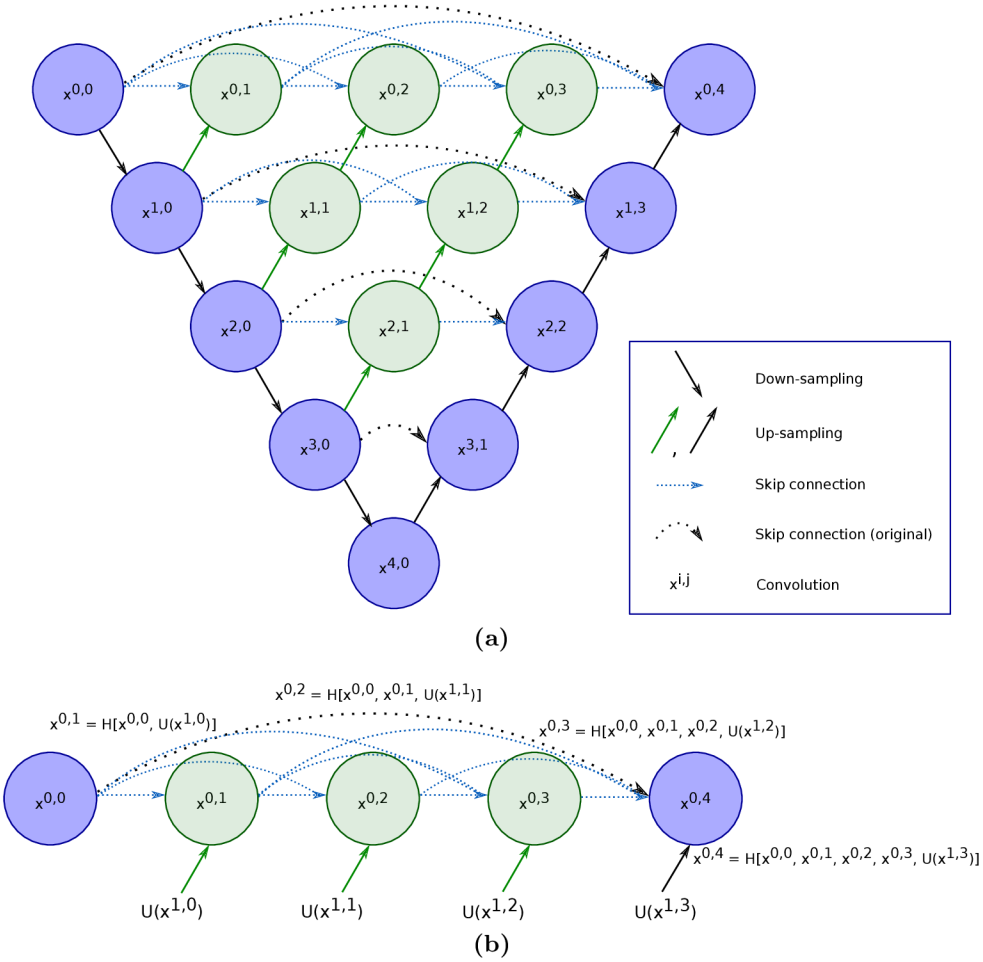


Figure 3.6: (a) A diagram of the Nested U-Net architecture. Purple indicates the Original U-Net structure. Green and blue components distinguish the Nested U-Net from the Original U-Net. The semantic gap between the encoder and decoder is bridged before the fusion. This can be seen, for instance, at the semantic gap between ($x^{0,0}$, $x^{1,3}$), which is bridged by a dense convolution block with three convolution layers. **(b) Detailed analysis of the first skip pathway.** Operation H represents the concatenation. Presented initially in [50].

Moreover, they propose to use deep supervision [20] in Nested U-Nets. It enables model pruning [4] that brings speed gain as it reduces network parameters. Thanks to the nested skip pathways, the full resolution feature maps are generated at multiple semantic levels: $x^{0,j}, j \in \{1, 2, 3, 4\}$. They are amenable to deep supervision. The model pruning process and the resulting performance after applying different pruning levels are in detail described in the paper [50].

3.3 Processing of 3D Data by Deep Neural Networks

Although the extension of deep neural network operations such as convolution from 2D to 3D domain seems natural, the additional computational complexity introduces notable challenges. Having volumetric data (for example, voxel models) or 3D surface data (for example, represented as polygon meshes) as an input to deep neural networks has a considerable drawback in computational time and memory requirements. With the advances in low-cost 3D acquisition devices, analyzing 3D shapes for tasks like classification, segmentation, or landmark detection became critical in many fields. A key technique is image feature extraction, where CNNs demonstrate their advantages. Several approaches were presented to address the non-trivial task of the appliance of CNN techniques on 3D models.

3.3.1 Voxel-based Approach

Voxel-based approaches model the input 3D data as a function sampled on voxels. Additionally, they define a 3D CNN over voxels for shape analysis. Wu *et al.* [48] introduce 3D ShapeNets – a CNN for object recognition and shape completion. This approach is limited to low resolutions due to the high memory and computational cost. A 3D voxel volume in this work is limited to resolution of 30^3, which has almost the same dimension as a 2D image of a resolution of 165×165. This method is full-voxel-based.

Rather than applying the CNN operations on the whole voxel volume, Graham *et al.* [13] propose the 3D sparse CNNs. These CNNs apply operations on active voxels only. This approach is efficient for architectures with a low number of convolution layers but becomes less efficient for deeper networks.

Built upon the octree representation of 3D shapes, Wang *et al.* [47] present Octree-based Convolutional Neural Networks (O-CNNs). The motivation behind their work is to decrease the computational time and memory requirements of 3D voxel volumes processing. The key idea is to present the 3D shapes with octrees and perform 3D CNN operations only on the sparse octants occupied by the boundary surfaces. They also design a novel octree structure that stores the features and associated octant information into the GPU memory to support all 3D CNN operations on GPU. Compared to the full-voxel-based approach, O-CNN can process voxel volumes of size 256^3 with significantly less memory occupation in much less time.

3.3.2 Manifold-based Approach

Manifold-based methods perform CNN operations over the features extracted from the geometry of a 3D mesh. Some of these methods convert the 3D surfaces into a geometry image so that standard CNNs can be directly used to learn 3D shapes [41]. Furthermore, a group of techniques called *Geometric deep learning* was introduced to generalize deep

learning to non-Euclidean domains such as graphs and manifolds. The concept of Geometric deep learning is explained in [5].

3.3.3 Multi-view Approach

An alternative way of 3D data processing by neural networks is a multi-view approach. Obtaining state-of-the-art results on 3D classification, Su *et al.* [43] presented the multi-view CNN idea. It is relatively straightforward and consists of three main steps:

1. Render a 3D shape into several images using varying camera extrinsics.
2. Extract features from each acquired view.
3. Process features from different viewpoints in a way suitable for a given task. In [43], a pooling layer followed by fully connected layers was used to get class predictions, as shown in Figure 3.7.

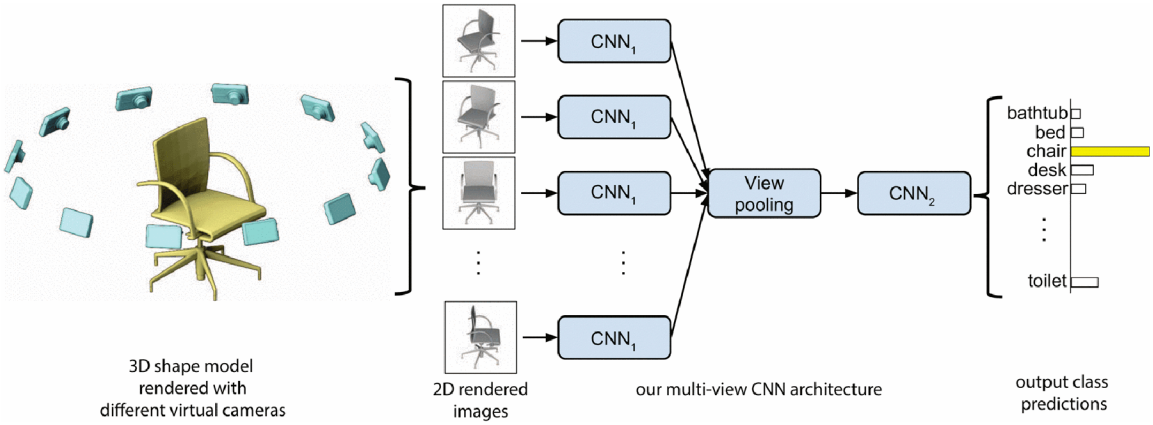


Figure 3.7: Outline of the method of multi-view 3D shape recognition. 3D shape is rendered from 12 different views and passed to CNN_1 modules. Outputs are pooled and passed through CNN_2 to obtain output class predictions. Adapted from [43].

Multi-view-based methods can process high-resolution 2D inputs, which can be rendered either from volumetric or surface 3D data. This study, however, leaves many questions unanswered. It is still unclear how to properly choose the viewpoint number and distribution, so it is always necessary to verify their fit for the given new task.

3.3.4 Applying a Multi-view Approach for Landmarking

Paulsen *et al.* [29] proposed a multi-view approach to identify feature points on facial surfaces. They have decided to use more views for image rendering, specifically 100. As the dataset is not purely medical, they chose a modified stacked hourglass architecture (see Figure 3.8) rather than the U-Net. They created several types of rendered images from each of the camera's positions to run their experiments. Besides the geometry rendering and texture surface rendering (RGB), OpenGL z-buffer was used to compute the depth map. Authors discussed the popularity of depth maps in machine learning, as it is often used as the input to the system, for example, when depth sensors are used to obtain data.

Additionally, the curvature was rendered and eventually combined with a depth map to scale down the localization error.

This paper's main contribution is the demonstration of the rendering pipeline, which offers an alternative way to handle complicated landmark placement on 3D surfaces. They discussed multiple geometry derivatives and experimented with their combinations to bring state-of-the-art results in feature point detection on facial 3D scans while decreasing the prohibitive GPU memory requirements needed for true 3D processing.

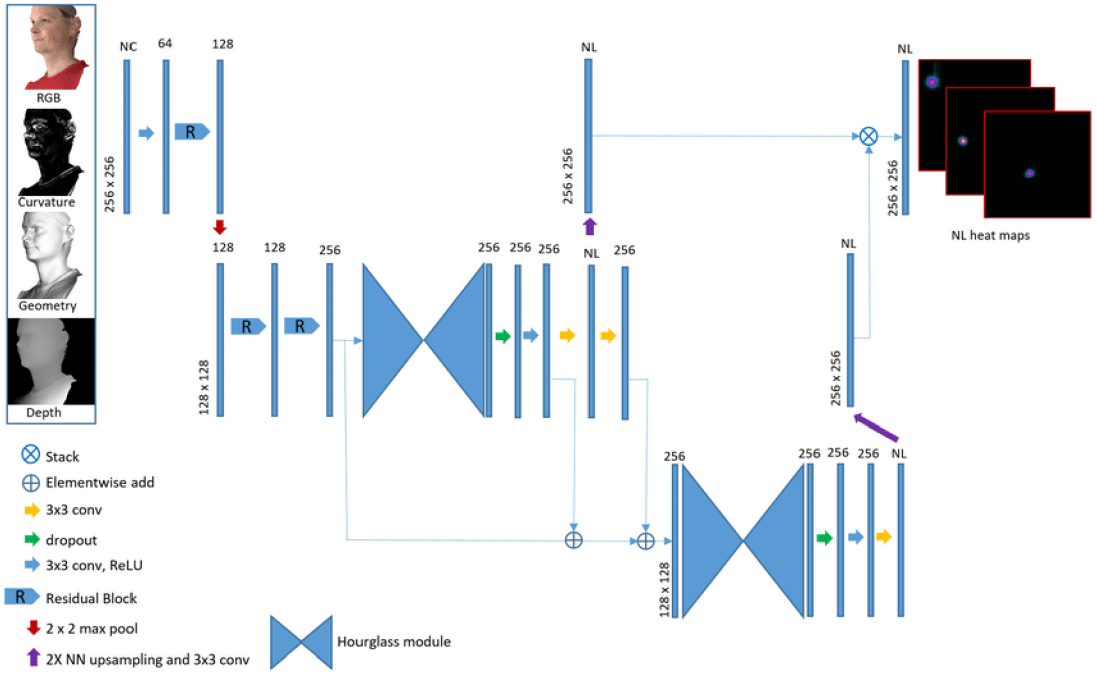


Figure 3.8: Network design for facial landmark localization presented in [29]. Notice that the input channels vary as the model works with different rendering configurations (RGB surface, curvature, geometry, or depth map). Adapted from [29].

Additionally, they proposed a consensus method to find the final estimate, which combines *least squares fit* and *RANdom SAmple Consensus* (RANSAC) [11]. For each landmark, N rays in 3-space are the outputs of the proposed method. To robustly estimate a 3D point from several potentially noisy rays, they defined each ray by its origin a_i and a unit direction vector n_i. Then, the sum of squared distances from a point p is calculated as follows:

$$\sum_i d_i^2 = \sum_i [(p - a_i)^T (p - a_i) - [(p - a_i)^T n_i]^2]. \tag{3.2}$$

It is necessary to differentiate this equation with respect to p. It brings the solution $p = S^+C$, where S^+ denotes the pseudo-inverse of S. In this case, $S = \sum_i (n_i n_i^T - I)$ and $C = \sum_i (n_i n_i^T - I)a_i$. RANSAC procedure initially estimates the value of p by three randomly chosen rays. The residual is computed as the sum of squared distances (see Equation 3.2) from p to the included rays, and the iterative RANSAC algorithm then performs I iterations. In each of these iterations, the number of *inliers* and *outliers* is calculated, respecting a predefined threshold τ. For further reading about the RANSAC algorithm, Section 4.5.2 describes its usage in this work. Even more details can be found in [11, 16].

Chapter 4

Proposed Solution for Orthodontics Landmark Detection on 3D Models

This chapter introduces the proposed solution for the detection of landmarks on the surface of polygonal models. At first, the task definition is presented. Afterward, the dataset of dentition used in this work is described. Later in this chapter, the overall method outline is introduced. The most essential concepts of the method are described in detail: the multi-view rendering pipeline, the design of proposed CNNs, and consensus methods.

4.1 Task Definition and Dataset

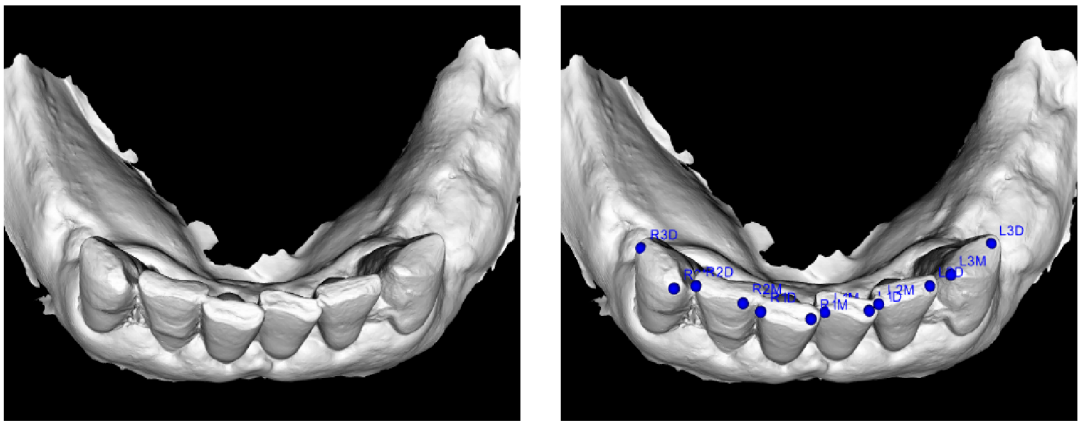
The main objective of this task is to automate the landmark placement in digital orthodontics. The goal is to detect points on specific surfaces of teeth (the desired positions of landmarks were discussed in Section 2.3). Figure 4.1 illustrates an example of an input surface model with corresponding annotated landmarks.

This task has two noteworthy and — in some way — contrary elements: (i) it is intended to be used as a part of a clinical application, thus, the computational time and memory requirements should be as low as possible, and (ii) the input data are represented in the form of 3D polygonal models. This representation yields significantly more information than a 2D image but has some drawbacks when processed in convolutional neural networks, as discussed in Section 3.3. Additionally, neural networks usually benefit from a large number of observed data. Limitations of the dataset of medical data must be taken into account.

I find it important to stress the difference between *keypoint*[1] and *landmark* detection, as their meanings are sometimes interchanged. Keypoint is an unlabeled point without any further meaning. It is described by its position in the world space[2] *only*. Landmarks have additional semantics. Each detected landmark is associated with a label. It always depends on the task that is solved, whether to use keypoint or landmark detection. See Figure 4.3 for an illustration of the distinction.

[1] also called *interest points* or *feature points*

[2] the world space position is a keypoint descriptor specific for the 3D scene, it might be expressed differently in other setups



(a) Example of an input polygonal model **(b)** Result of the landmark detection

Figure 4.1: Illustration of an input and corresponding output. The method loads a surface model and finds the positions of the orthodontics landmarks, as suggested in Section 2.3.

4.1.1 Proposed Multi-view Landmark Detection Approach

To accomplish the goal of this task, a method that combines multiple state-of-the-art approaches is presented. This method works out the high computational time of 3D data CNN processing and data shortage on several levels:

1. **Simplification of the task of 3D landmark detection by moving to 2D space**

   2D images are rendered and used per consequens as inputs to neural networks to decrease the computational time. Networks are trained and evaluated on depth maps rather than on model geometry.

2. **Regressing heatmaps rather than (x, y) coordinates directly**

   This approach is widely used in both medical and non-medical fields. Its benefits are elaborated in Section 3.1.

3. **Usage of model designs that respect dataset limitation**

   The U-Net architecture (see Section 3.2) was designed to bring satisfying results even on small medical datasets. Its offshoots are used to regress heatmaps with a Gaussian placed on ground truth landmark positions.

4. **Making use of 3D information by applying the multi-view approach**

   Dropping the third dimension data completely might be highly disadvantageous. The model is observed in the scene from multiple views. In comparison to a single view, this method should bring more accurate predictions. A similar approach to the one discussed in Section 3.3.3 is presented.

5. **Finding a consensus method that predicts the positions with the highest accuracy**

   Multi-view approach comes with an additional task. If the model is observed from N views, there are logically N predicted positions. It is necessary to find a final estimate by applying a consensus on acquired predictions.

4.1.2 Dataset

In this work, 491 polygonal models of human dentition in STL[3] format were provided.[4] Both maxillary and mandibular dentition occur within the dataset. These dentition scans were obtained from different patients, bringing significant variation among the models, primarily due to the frequent teeth missing. Neither public datasets of similar 3D models of dentition nor the landmark ground truths are available. The ground truth annotation was done by myself with a custom annotation tool, which will be presented in Section 5.3. The process of landmarking was done without any professional orthodontist supervision, but it was shown to me by the thesis supervisor.

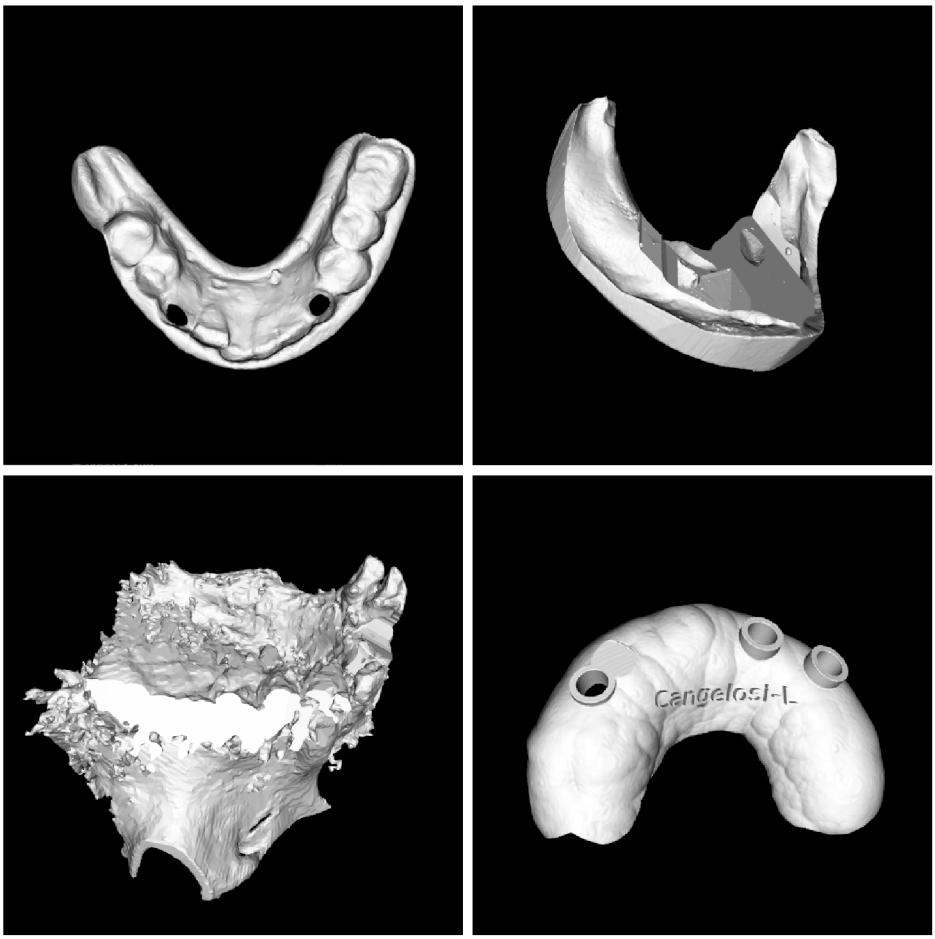


Figure 4.2: Typical cases of discarded models. Discarded polygons are typically castings of teeth or contain significant noise from the scanning process. Such deflexions would confuse the network as they have no teeth surfaces for proper landmark placement.

Some of the provided models were discarded as the dentition was partially or entirely missing (see Figure 4.2 for examples), resulting in **269 polygonal models** suitable for training, evaluation, and testing.

Those models, however, are not in a feasible format to serve as an input to the proposed method. The method is designed to process 2D data, as discussed in Section 4.1. With

[3] https://docs.fileformat.com/cad/stl/

[4] The dataset used in this work was provided by TESCAN 3DIM, s.r.o.

the custom annotation tool, a dataset of depth maps with corresponding ground truths is created. In terms of dataset preparation, the multi-view rendering pipeline serves as an additional augmentation tool that significantly increases the dataset. With the number of views set to 100 and with 269 polygonal models, **26 900 depth maps** are available to train and evaluate the system.

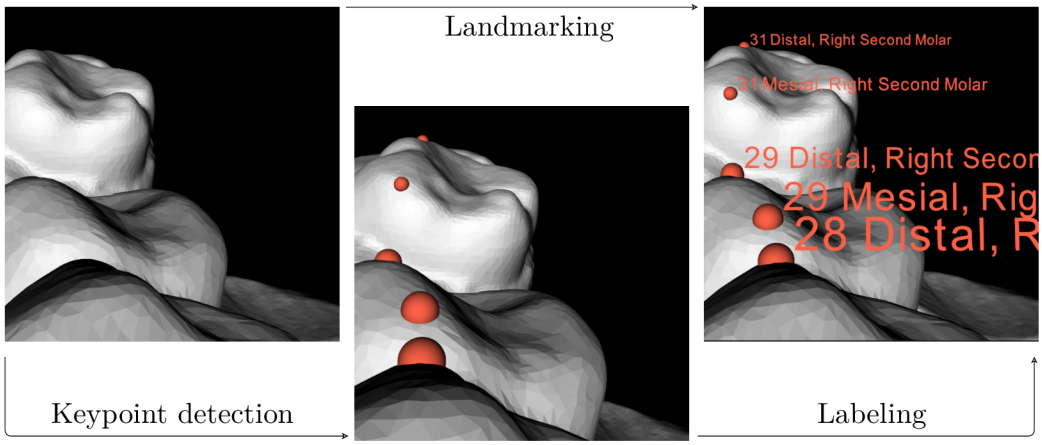


Figure 4.3: Difference between keypoint and landmark detection. The landmarking process can be sub-split into keypoint detection and labeling. In this work, landmarks are detected, as their label is vital for further processing.

4.2 Method Outline

The detailed outline of the proposed method is depicted in Figure 4.4. The rendering pipeline (Section 4.3) is used to obtain N depth maps from N different views. The number of views is chosen experimentally and serves as a possible *hyperparameter* for experimenting with the system. To illustrate — as discussed in Section 3.3 — in the work of Su *et al.* [43], 12 views were used. Furthermore, in [29], multiple experiments with multiple (25, 50, 75, and 100) N values were conducted.

The predictions from the network are 32 heatmaps representing the coordinate estimates for each landmark. It is necessary to calculate a display coordinate (x, y) in $\mathbb{R}^2$ from each of the heatmaps. Assuming the highest confidence in the point where the maximum value occurs, the *Non-Maximum Suppression* (NMS) algorithm is used to find that position. The aforementioned (x, y) values are the landmark's display coordinates, as the network is trained to regress the converted world coordinates to display coordinates (Section 5.3).

The (x, y) display coordinates of one view are not the final outcome. It is indispensable to propagate the information into a world coordinate system $\mathbb{R}^3$ and find a final estimate by combining outputs from several camera views.

The information propagation to $\mathbb{R}^3$ is done by coordinate conversion. With the known position of the center of projection, the prediction for a single view of one landmark can be interpreted as (i) **a ray** defined by the origin in the corresponding center of projection and the point on the view plane at detected display coordinates or (ii) simply **a point** in the 3D scene, i.e., the converted display coordinate into 3-space.

However, this still does not cover the multi-view approach. Two consensus methods are used and presented in Section 4.5, one for each of the aforementioned interpretations. These methods find the estimation among multiple predictions. The last necessary step is to find

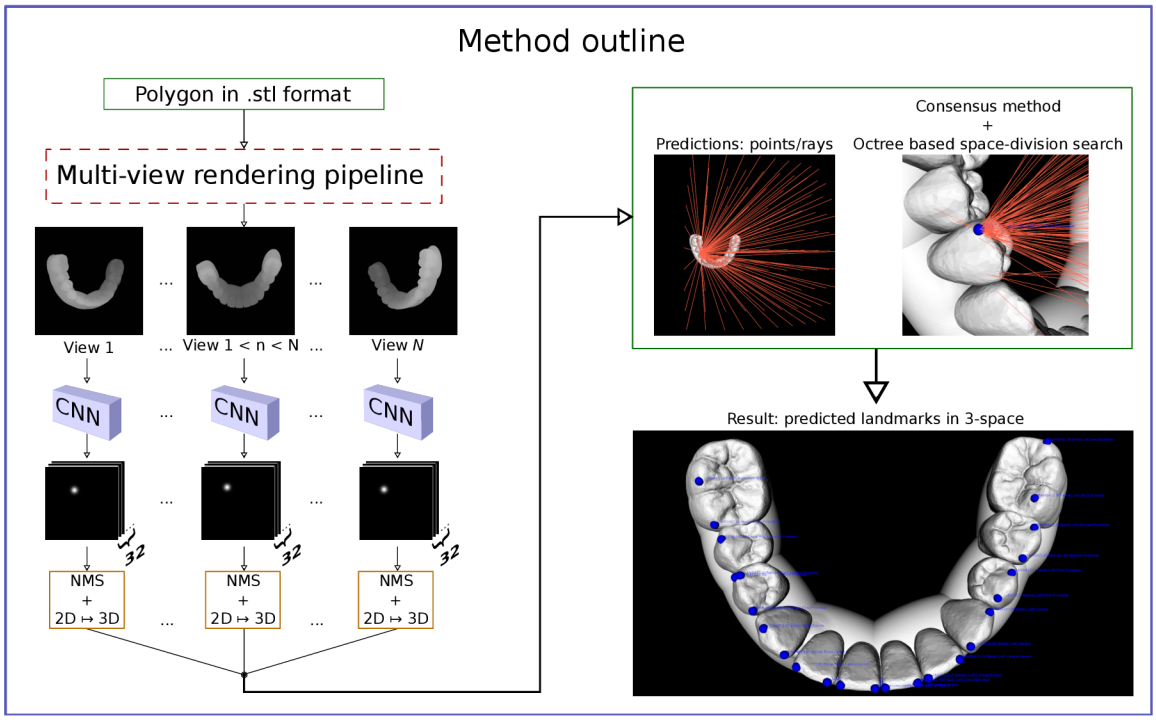


Figure 4.4: Overall method outline. The evaluated 3D model serves as an input to the multi-view rendering pipeline. According to the number of views, a corresponding number of depth maps is evaluated by CNN modules. The extracted display coordinates of landmarks in each view are used in the consensus method to get a single estimate.

the closest point on the surface of the polygonal model, as the consensus output does not guarantee the placement on the model surface. An octree data structure contains a recursively sub-divided target polygonal model. Each node stores an explicit 3D point, which is the center of a given subdivision. In the leaf nodes, individual surfaces of the polygonal model are stored. With such representation, the octree-based space-division search algorithm can be used to estimate the final output. The closest point on the surface of the polygonal model to the consensus output is considered to be the final estimate.

4.3 Multi-view Rendering Pipeline

I defined the task as a 2D regression of heatmaps (see Section 3.1) that respects the multi-view approach. To accomplish that, it is necessary to render the 3D object into a suitable format. I have decided not to render the object geometry directly but to use the given object's depth maps instead. From the view of a human, object geometry contains more information about the rendered model – it is more obvious where to place a landmark. This is not necessarily the case of neural networks – they are able to obtain more information from the depth value in each pixel. The type of rendering might vary according to the task that is solved. A similar approach was chosen in the work described in Section 3.3.4, but they used other geometry derivatives as well.

To meet the goal of multiple depth map rendering, a multi-view rendering pipeline was designed. It is depicted in Figure 4.5. The rendering pipeline is used in the annotation tool as well as during the evaluation of unseen polygons.

The model is loaded into the scene and must be placed within the viewing frustum in a position that meets the needs of the solved task. Afterward, the model is observed from several views. Different observations are obtained by moving the camera in the scene. Finally, the scene is rendered in each of the views. The number of observations should always be cross-validated for the given task. The same applies for the camera positions.

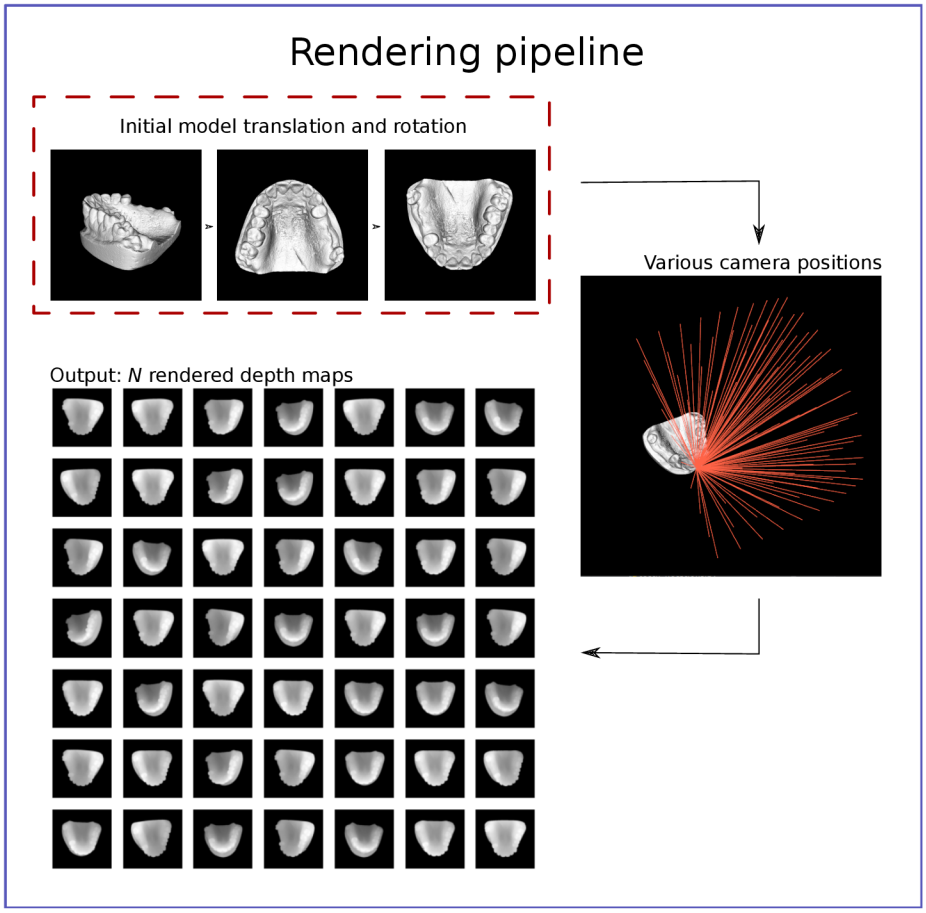


Figure 4.5: Outline of the multi-view rendering pipeline. After the initial application of affine transformations, the z-buffer is used to calculate the depth value of each pixel of the observed scene to form a depth map. Furthermore, this depth map rendering is done with multiple (N) camera extrinsics to follow the multi-view approach. Therefore, the output of the rendering pipeline is N depth maps from N different views.

4.4 Proposed Designs of CNN Models

To get the landmark predictions, three neural network architectures were trained. Each of these networks is trained on the depth maps generated by the proposed annotation tool. The network does not distinguish between mandibular and maxillary dentition. In the latter case, the model is rotated to be in the requisite position. The first proposed network is a **BatchNorm U-Net**. This network has a similar design as the original U-Net, which was discussed in Section 3.2. I applied additional batch normalization layers between the convolutional and ReLU activation layer. Additionally, the input size to the network is 128×128, as I was limited by the computational power. I used padding to prevent the loss

of border pixels. Thus, the exact dimensions (not the size of features) as the input image are produced. The number of output features is changed, so it corresponds to the number of detected landmarks. Figure 4.6 shows the modifications applied to the original U-Net.

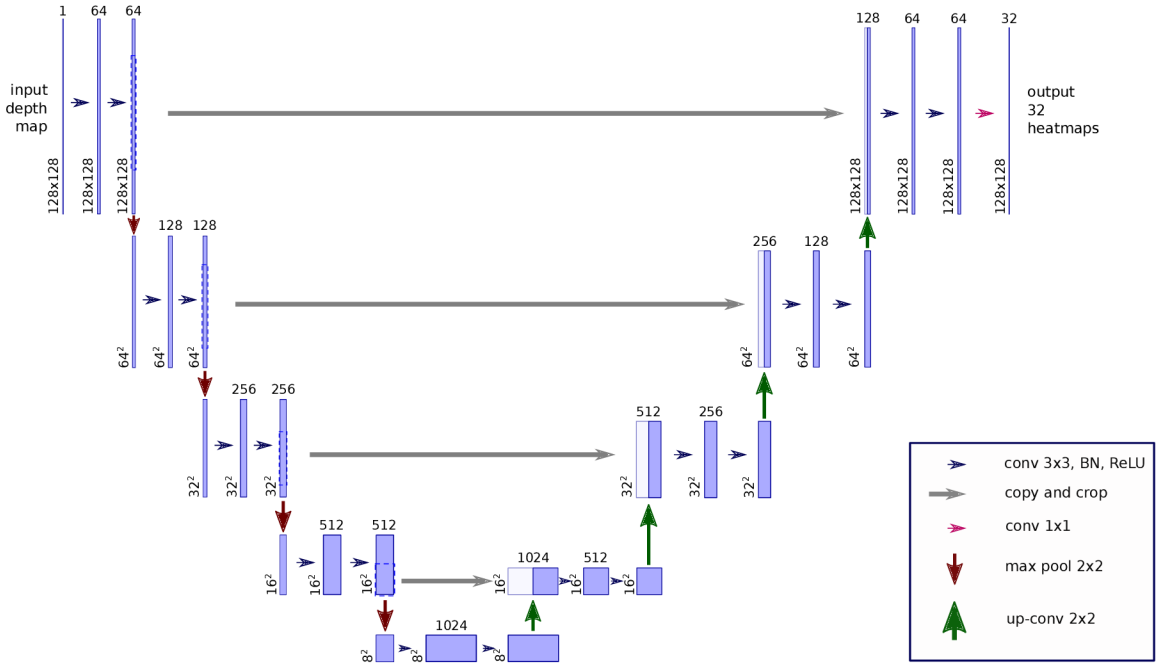


Figure 4.6: BatchNorm U-Net used in this work. Comparing with the original U-Net [35], the dimensions of the input image are changed. Padding is used to produce feature maps of the exact dimensions as the input image. Batch normalization is inserted between the convolution and activation layers to speed up the training and reduce generalization error.

The second trained network is the **Attention U-Net**, which integrates the *Attention gates* into the original U-Net. Detailed descriptions, as well as the architecture design, can be found in Section 3.2.3. The input image size is 128 × 128, and the number of output feature channels is 32.

The third trained network is the **Nested U-Net**. This network design is described in Section 3.2.4. The input size is again set to 128 × 128, and the number of output channels is 32. I assume that reducing the semantic gap between the encoder and the sub-networks brings more accurate predictions, even on the heatmap regression task. No network pruning was integrated. The architecture of the Attention U-Net and the Nested U-Net was desgined according to the original papers [28, 50], and the implementation is also inspired by [2], where authors shared their source files online.[5]

4.5 Consensus Methods

One of the critical steps in a multi-view approach is to find a final estimate from all calculated predictions. In this work, the prediction can be interpreted either as a point in the world coordinate system or as a ray passing through the center of projection and the

[5] https://github.com/bigmb/Unet-Segmentation-Pytorch-Nest-of-Unets

predicted display coordinate. As Figure 4.7 indicates, point L represents the point interpretation, and $\overrightarrow{CL'}$ represents the ray interpretation in the world coordinate system. To compere these two representations and how they are used in consensus methods, see Figure 4.9.

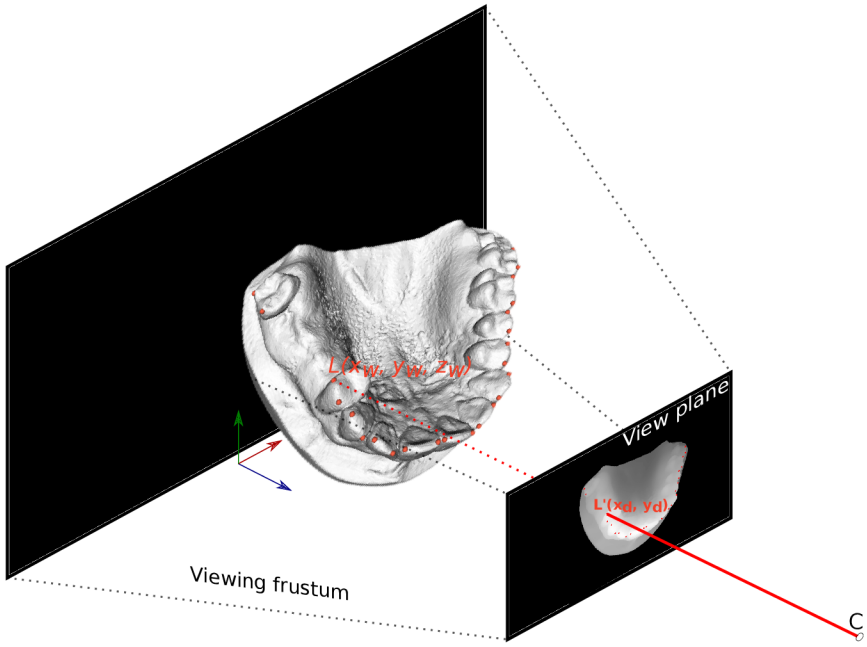


Figure 4.7: The depiction of a polygonal model in a scene. The model is placed within the viewing frustum. L stands for a point representing a randomly selected keypoint on a polygonal model, located at x_w, y_w, z_w within the world coordinate system $\mathbb{R}^3$. When the depth map is rendered, point L' is calculated. It represents the corresponding point position within the display coordinates x_d, y_d in $\mathbb{R}^2$. Point C represents the center of projection.

4.5.1 Centroid of Points in 3D space

Let's consider N as the number of views used in the multi-view approach. Let's also interpret the single-view evaluation output as a point on the target polygonal model. With N views, the consensus output P is a single point in Euclidean 3-space $\mathbb{R}^3$ and is calculated from N points in $\mathbb{R}^3$ as follows:

$$\mathrm{P} = [\frac{\sum_{i=1}^{N} x_i}{N}, \frac{\sum_{i=1}^{N} y_i}{N}, \frac{\sum_{i=1}^{N} z_i}{N}] \tag{4.1}$$

where x_i, y_i and z_i are the (x, y, z) coordinates of ith output. Note that this point is not the final estimate because it is not automatically located on the surface of the target polygonal model.

4.5.2 Random Sample Consensus

To suppress the potential accuracy loss brought by the presence of outliers, the *RANdom SAmple Consensus* (RANSAC) comes forward. The algorithm was initially proposed by Fischler and Bolles [11], and it is a general parameter estimation approach designed

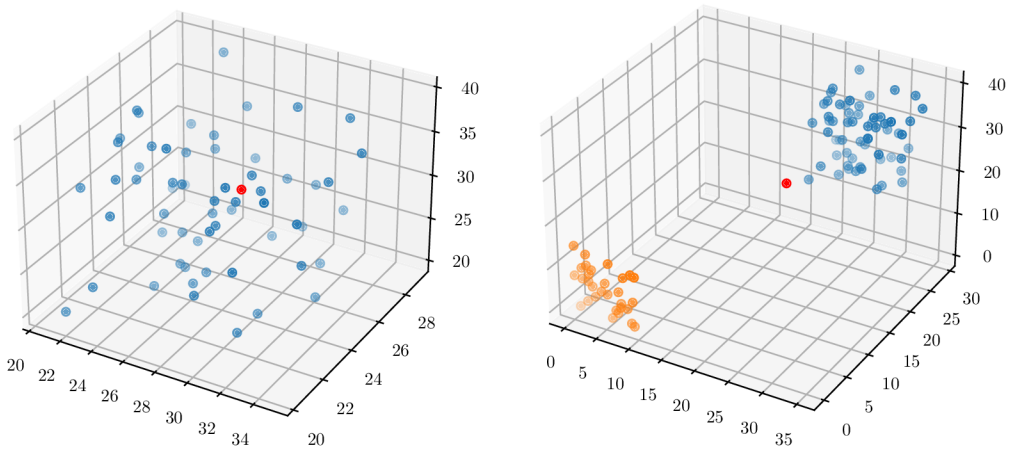


Figure 4.8: Depiction of the Centroid value without and with outliers. The red point represents point P from Equation 4.1, blue and orange points represent the input points. The left picture shows the ideal situation, where all points follow one distribution. In the right picture, two different distributions were used to generate the points. Orange points present potential outliers produced by some of the predictions. The result is highly affected by them.

to cope with a large proportion of outliers in the input data. The final estimate of each landmark could be found as an intersection of corresponding rays in $\mathbb{R}^3$.

As the rays are (post-processed) estimates from a neural network, it is necessary to consider that it predicts the outputs with uncertainty. Then, the benefits of RANSAC in this task can be twofold: (i) The consensus estimate of one point out of N predictions and (ii) the division of predictions into *inliers* and *outliers*.

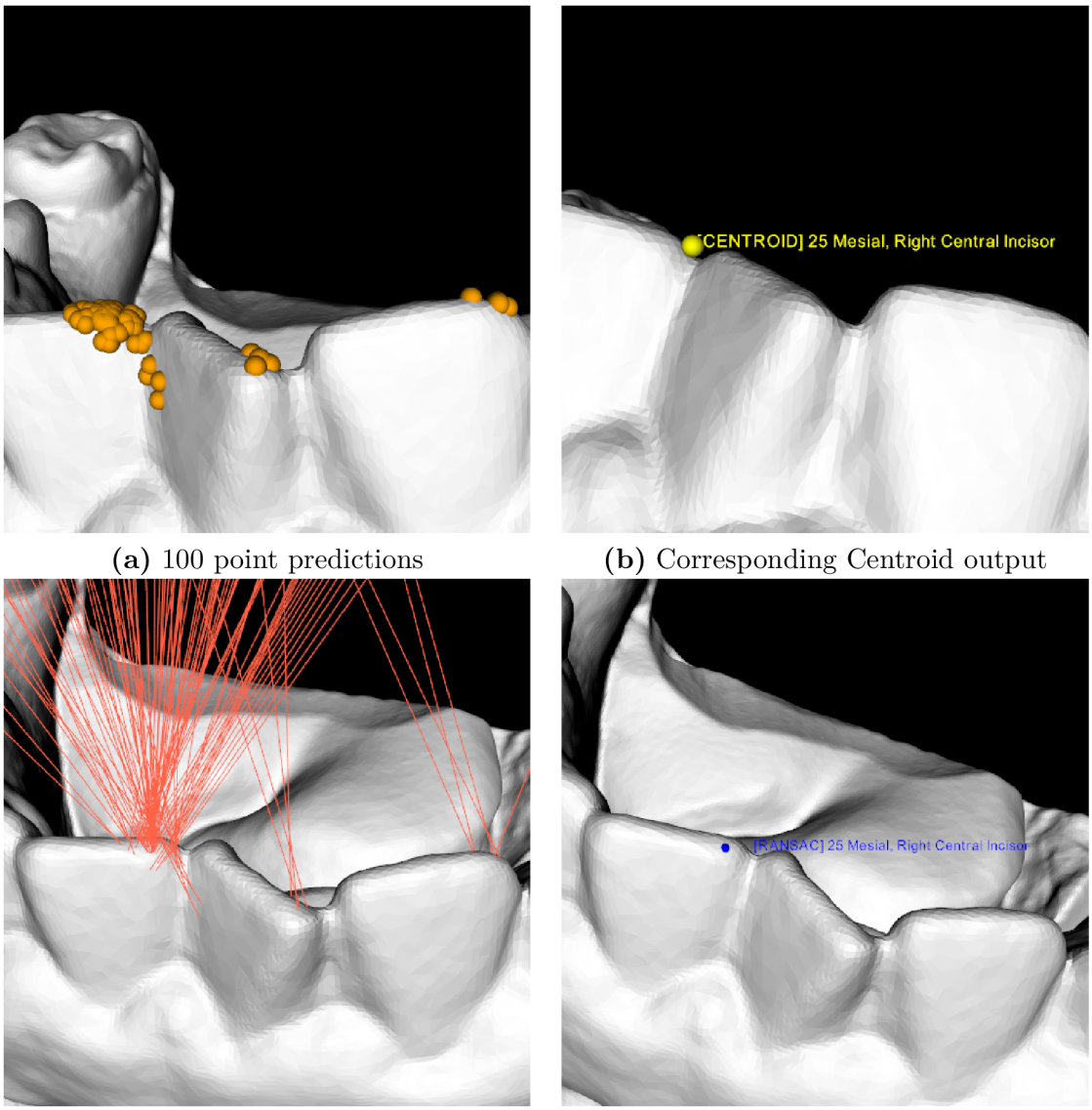
As for the consensus estimate, the RANSAC algorithm is combined with *least-squares fit* (LSQ) in a similar way as in [29]. Their approach is summarized in Section 3.3.4. The algorithm is as follows:

Algorithm 1: RANSAC and LSQ to find the landmark estimate from N rays in $\mathbb{R}^3$

Result: A point estimate in 3-space

1. Select randomly n rays, where $n < N$;
2. Find the initial estimate of point P by finding the intersection point of selected rays in 3-space, in the least-squares sense;
3. Compute the sums of squared distances from all rays and point P;
4. Determine how many rays from the set of all rays fit with the predefined tolerance ϵ, i.e., determine the number of inliers;
5. If the fraction of the number of inliers over N exceeds a predefined threshold τ, re-estimate the point P using all the identified inliers. Compute the sum of squared distances from all rays and re-estimated point P. If the sum is lower than the best sum, this sum is now the best one and P is considered as the best estimate;
6. Repeat steps 1 through 5, a maximum of I times.

The number of views N in this work is various, namely 9, 25, and 100. Number n is chosen to be 3. The tolerance ϵ is 5 mm – if the distance from a ray to point P is less than 5 mm, such a ray is classified as an inlier. Threshold τ is set to be one-quarter of N. The algorithm stops after 50 iterations, i.e., $I = 50$.



(a) 100 point predictions

(b) Corresponding Centroid output

(c) 100 ray predictions

(d) Corresponding RANSAC output

Figure 4.9: Centroid and RANSAC as consensus methods. Picture **(a)** shows 100 predicted point positions for the mesial landmark of tooth 25. **(b)** shows the result of the calculation of Centroid. **(c)** shows the 100 predictions interpreted as rays. Finally, **(d)** depicts the output of the RANSAC consensus method. Speaking of both outputs, it is not located on the polygon surface directly as the last step of the pipeline has not been applied yet.

Chapter 5

Implementation

This chapter introduces the technologies used for the implementation of the proposed method. Later in this chapter, the details about the implementation of the rendering pipeline and the annotation tool are described.

5.1 Technologies

I have chosen Python as a programming language. The main reason for choosing Python is its simplicity, which allows me to focus on the application logic. Another reason is the availability of packages and frameworks for machine learning tasks. Two major Python libraries were used to implement this work. The Visualisation Toolkit[1] is essential for the depth map rendering, multi-view rendering pipeline, coordinate propagation, and landmark placement on the surface of the target model. PyTorch[2] is used for the training and evaluation of the proposed neural networks.

The Visualisation Toolkit

The Visualisation Toolkit (VTK) is open-source software for scientific data manipulation and visualization, 3D graphics, image rendering, and more. VTK is implemented in C++, but as it allows binding to other languages, for example, Python, the performance of C++ is acquired while writing a simple syntax Python code.

PyTorch

PyTorch is a Python deep learning framework. It provides two key features:

- Replacement of NumPy[3] operations with operations accelerated by GPU,
- API for deep neural networks construction.

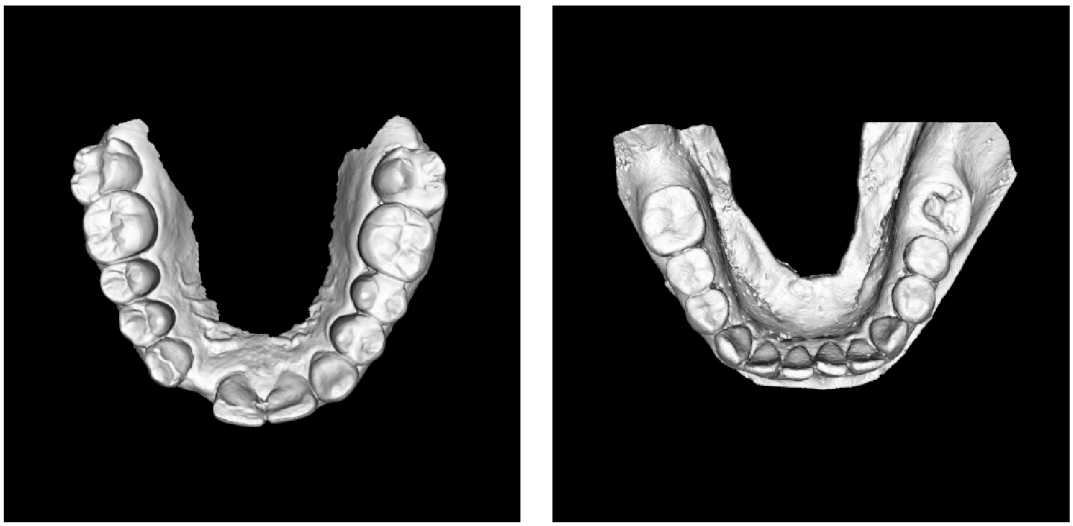
This framework is leaning towards Python in the words of simplicity and used concepts. Its ease of use surfaces from dynamic computation and straightforward syntax. Dynamic computation brings immense flexibility, especially when complex architectures are constructed. Concepts like classes and structures are used extensively, similar to those

[1] https://vtk.org/
[2] https://pytorch.org/
[3] https://numpy.org/

in Python. Unlike other deep learning frameworks, PyTorch allows the building of deep neural networks in a pure object-oriented paradigm without bringing its own programming techniques [44].

5.2 Rendering Pipeline Configuration

The first step of the rendering pipeline (Section 4.3) is the object loading into the scene. At first, its center of mass is calculated, and the model is translated as close to the origin of the world coordinate system as possible. Furthermore, the method expects to have two prior information about the rendered model: (i) the dentition type (maxillary or mandibular) and (ii) the transformation matrix provided in corresponding XML[4] file. These two pieces of information are used to create a composed affine transformation applied to the rendered polygonal model. However, after some observations of the dataset, there are some exceptions when the meshes are not always rotated correctly. For that reason, some manual manipulation of the camera is allowed, such as zooming, rotating, and panning, to ensure the model and camera are suitable for depth map rendering. The possibility of manual camera settings actually makes the aforementioned prior information unnecessary, making the pipeline general and not specific for the dataset used in this work. However, the transformations are applied in this work anyway, just to eliminate the time needed for manual interaction. The requisite position of the polygonal model is described and depicted in Figure 5.1.



(a) Maxillary dentition **(b)** Mandibular dentition

Figure 5.1: Requisite position of both dentition types. Pictures **(a)** and **(b)** show the ideal position of the polygonal model in the 3D scene. The goal is to face the occlusal and incisal surfaces towards the camera. Additionally, the maxillary dentition is rotated to be in the same position as the mandibular dentition. This i.a. means that from the perspective of network training and evaluation, the dentition type is not distinguished.

The near and far planes are set so the model is in the *viewing frustum*. Afterward, the camera is set to N different positions. The camera positions are not set randomly, but they

[4] https://www.w3schools.com/xml/xml_whatis.asp

follow a particular pattern. The first rendered image is obtained with the camera's azimuth and elevation, both set to $-30°$. The camera's azimuth and elevation are then iteratively modified, ending with the azimuth's values and elevation, both set to $+30°$ concerning the initial value. I find it unnecessary to set the camera to positions where the side or the model's back would be observed. This approach is appropriate for tasks like classification or shape synthesis, where the features acquired from such views are valuable [42, 43]. The rays representing each of the views form geometry similar to *pyramid*. A similar approach was chosen in [29], with the difference of variant camera position settings. Finally, at each of the camera's positions, the depth map is rendered. The OpenGL z-buffer is used to compute this distance map.

5.3 Annotation Tool

As I described in Section 4.1.2, the polygonal models were provided unannotated. The precision of annotations is essential for the overall results of the system. It is a highly time-consuming and repetitive task. A custom annotation tool that suits the particular needs of the task was designed and developed, its outline is depicted in Figure 5.2.

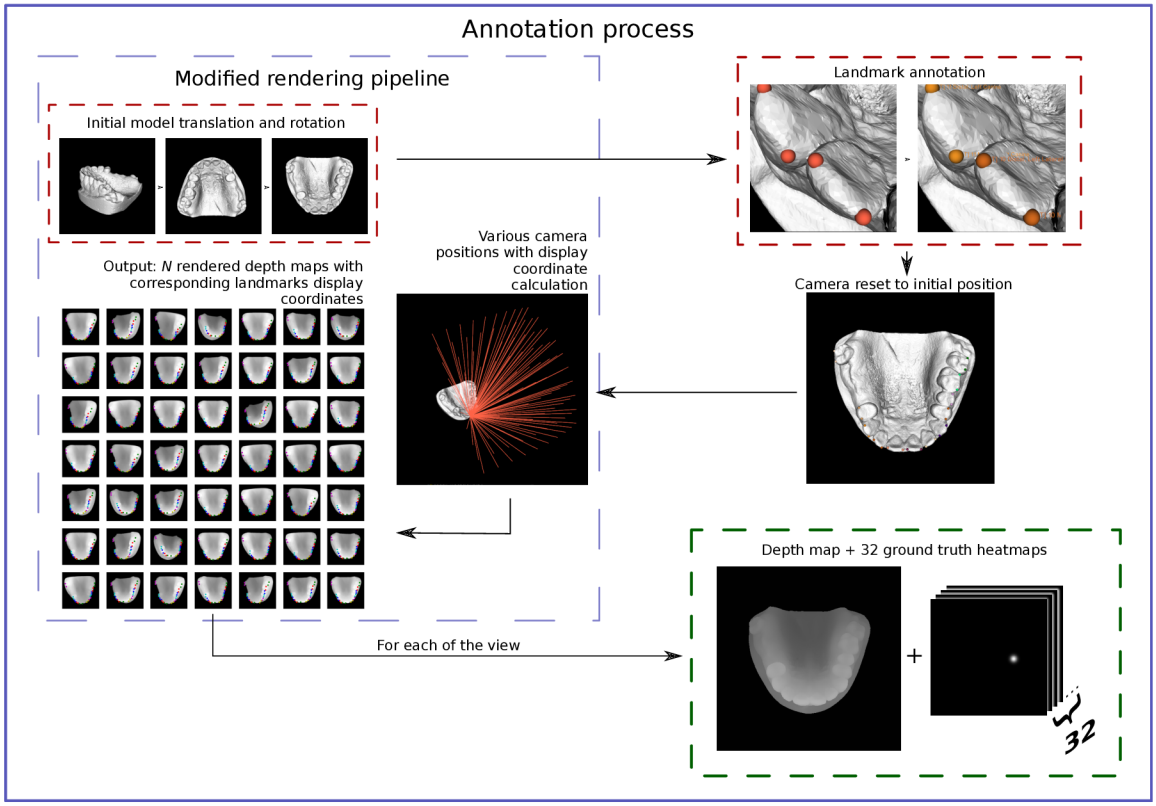


Figure 5.2: Annotation process. It starts with the first step of a rendering pipeline – the initial model transformation. Afterward, the manual camera manipulation is available, and the annotator creates the landmarks annotations in two steps – keypoint placement and labeling. The camera is set to the initial position, and the steps from the rendering pipeline then continue, enriched by the calculation of all landmarks' display coordinates. In the last step, for each of the views, ground truth heatmaps are created by applying a 2D Gaussian filter with the peak in each of the landmarks' positions.

The core of the annotation tool is the multi-view rendering pipeline. It contains an additional stage, during which the annotator is allowed to add keypoints on the surface of the dentition model and eventually label them with appurtenant tooth notation. The whole annotation process takes place in the 3D scene before the multi-view depth map rendering. This brings high annotation accuracy, as the camera operations like zooming, rotating, and panning allow the annotator to observe the model surface at close quarters. The process is designed to annotate keypoints first and label them in a separate step. This separation of landmarking process makes the tool more general, as the keypoint placement part is the same among all tasks. If additional keypoint semantics is demanded, the labeling part is easily modifiable. The annotated positions are exported in a form of csv[5] files.

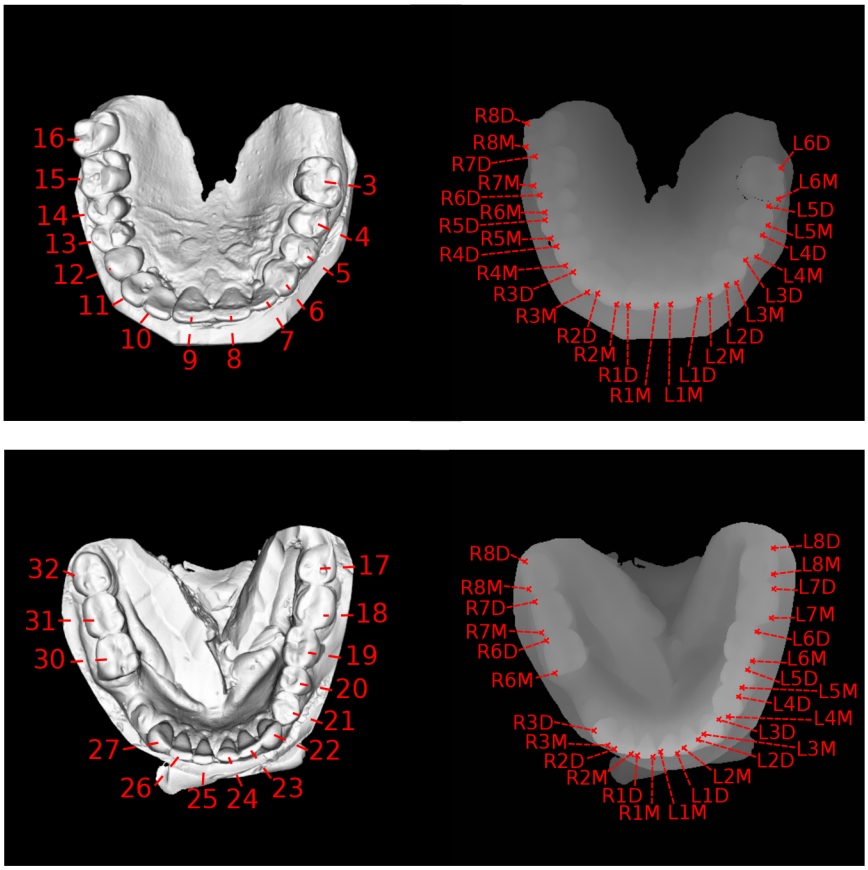


Figure 5.3: Teeth and landmarks notation. This figure shows the difference between a tooth and a landmark notation. The left parts of the pictures show the notation of the teeth on maxillary and mandibular dentition, respectively. The maxillary dentition is rotated, so it is in a requisite position. The landmark notations are the same for both the maxilla and the mandible. This means that the network does not distinguish between these two. The landmark notation is composed of three characters: the first one represents the Left or Right part, the second describes the tooth number within the quadrant, and the third one says whether the given landmark is on the Mesial or Distal surface.

Keypoints and landmarks in the scene are represented in the form of geometric primitives (spheres). Keypoints are placed in the scene by keyboard interaction. When the key P is

[5] https://docs.fileformat.com/spreadsheet/csv/

pressed, a ray is cast in the scene through the display coordinate at the mouse position. Keypoint is placed on the coordinate of intersection of the ray with the annotated 3D model. The labeling is done with the keyboard interaction as well. Landmarks $1 - 8$ are labeled with the keys 1 to 8, with the need of switching between the left and right side of the dentition. Labeling of landmarks follows the notation depicted in Figure 5.3. By pressing the key D, a misplaced keypoint or landmark is deleted.

The annotation prior to the depth maps rendering has a substantial advantage – the annotation is done just once for 100 depth maps. This is possible thanks to the **conversion of the world coordinates in** $\mathbb{R}^3$ of all landmarks **into the display coordinates** $\mathbb{R}^2$ at each of the camera's positions. In other words, at each of the camera's positions, a tuple (X, y) is created, where X stands for the calculated depth map and y are the values of display coordinates of k landmarks.

Additionally, the generated tuple (X, y) is modified to a tuple (X, y_h), where X is the unchanged depth map, and y_h is an $n \times n \times 32$ volume of heatmaps, where n is the depth map and ground truth heatmap dimension. Each of the heatmaps is generated during an additional step, where a 2D Gaussian filter with the variance of 10 and the amplitude of 1 is used to convolve the original screen landmark coordinates. This ensures that the network regresses heatmaps rather than the (x, y) coordinates directly, as defined in Section 4.1.

Chapter 6

Experiments and Results

The main interest of this chapter is to describe the conducted experiments and present their results. To find the best-performing method configuration, the following method parts are validated and compared:

1. **Different network architectures**

   Experiments test the proposed method with three trained architectures described in Section 4.4: **BatchNorm U-Net**, **Attention U-Net**, and **Nested U-Net**. I expected the integration of attention gates and the dense skip pathways to bring more accurate results than BatchNorm U-Net.

2. **Consensus methods**

   Besides the landmarking accuracy, the experiments conclude which consensus method should be preferred when the multi-view CNN approach is used. I compare the performance of the Centroid consensus method (Section 4.5.1) with the performance of the geometric approach (Section 4.5.2).

3. **Number of views used for the multi-view approach**

   The number of views is still an unclear parameter of the multi-view approach, as discussed in Section 3.3.3. I have chosen three numbers of views – 9, 25, and 100. I hypothesize that the accuracy should increase with the increasing number of views.

Results of a set of experiments are presented in this chapter. This set contains the comparisons of the performance of the overall method with all possible combinations of the aforementioned architectures, consensus methods, and the numbers of views. The goal of the experiments is to find the best-performing combination of these configurations. It also analyses the comparison of consensus methods and whether the increase in the number of views really brings better results.

All metrics are measured in physical units (mm) since the end clinical application is related to physical units. In terms of 3D landmark detection, no prior experiments were conducted on polygonal models from this dataset. All results are measured as an average of five evaluations as the model positioning in the requisite position slightly affects the results. Only positive patterns that are correctly classified are measured (for example, false positive landmark placement is not taken into account in accuracy measurements). Descriptions of all metrics used in this chapter can be found in Appendix A.

6.1 Training Procedure

The input to the proposed neural networks is a single channel depth map of size 128×128. This size was chosen as a compromise between acceptable training times and preserved image information. The training procedure ran on an NVIDIA GeForce RTX 2060 with 6 GB of memory.

6.1.1 Data Split and Augmentation

The polygonal models were divided into two groups – those used in the training procedure and those used for the evaluation. From the 269 valid polygonal models, 208 models (20 800 depth maps) were used for the network training, and 61 models were used to evaluate the proposed method. Furthermore, the 20 800 depth maps were split in the ratio of 4:1 into a training set and validation set, respectively.

Although the rendering pipeline brings one form of augmentation (see Section 4.1.2 and Section 4.3), some other augmentations were applied on the training and validation depth maps and ground truth heatmaps:

- **Scale** from the range $[0.90,\ 1.10]$,
- **Rotation** from the range $[-11.25,\ 11.25]$ degrees,
- **Translation** from the range $[-10\,\text{px},\ 10\,\text{px}]$ and applied in both vertical and horizontal directions.

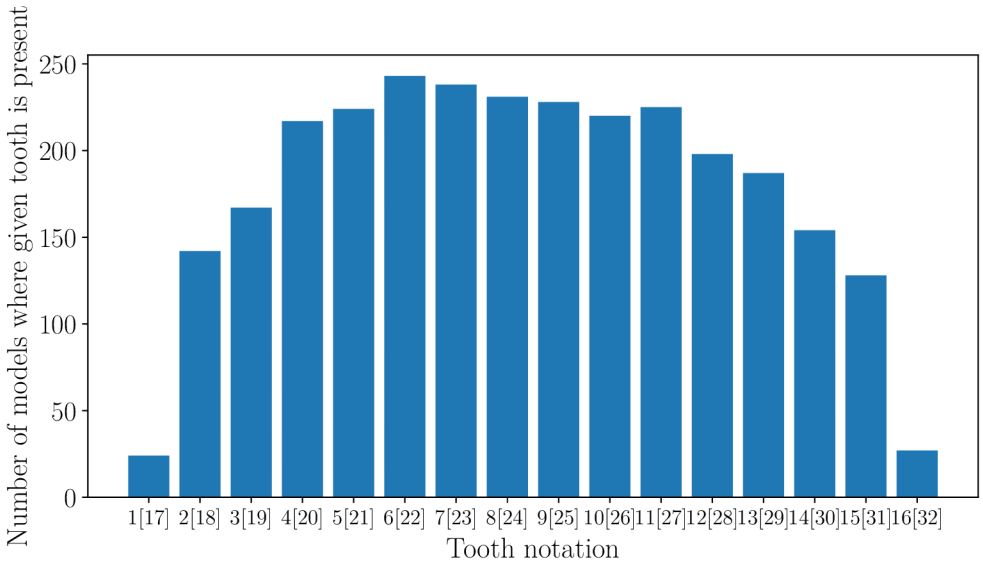


Figure 6.1: Graph describes the teeth presence in polygonal models. With the total number of 269 suitable models, the imbalance in terms of the teeth missing is significant. Teeth 1 (17) and 16 (32) are present in less than 10% of the polygons used for training. On the other hand, canines and incisors are present in the vast majority of models. Note that teeth 1 and 17 are considered the same, likewise to the rest of the teeth.

6.1.2 Training Parameters and Loss Function

Networks are trained using the Adam optimizer with the weight decay set to 10^{-3}. These specifics were chosen to keep the weights small and avoid exploding gradient, leading to reduced overfitting. The learning rate is initially set to 10^{-3}. Its value is dynamically reduced using **learning rate scheduler**.[1] The scheduler is used to get out of a loss plateau caused by reaching saddle points or local minima. The learning rate is reduced by a factor of 0.5 every time the value of validation loss has not improved for 5 consecutive epochs. The validation loss is monitored for the **early stopping**[2] as well. If the validation loss value does not improve for more than 30 consecutive epochs, the training is stopped. To reduce the memory requirements during training, the **automatic mixed precision** was used. It tries to match each operation with its appropriate data type. Operations like linear layers are much faster in `float16`. Other operations, on the other hand, like reductions, require the dynamic range of `float32`. The autocast[3] and GradScaler[4] are used to accomplish the mixed precision. The batch size is set to 32. With this size of the batch, I was able to train solely the BatchNorm U-Net. To train the Attention U-Net and Nested U-Net, I applied **gradient accumulation**. These models require more memory, which caused out of memory issues. The accumulation keeps adding gradients of the parameters for 4 number of batches. The batch size is set to 8, and then after 4 batch iterations, the updates are applied using the average of all gradients accumulated over 4 iterations.

To train the models on a regression problem, which is the case of this task, the **Root Mean Square Error** (RMSE) loss was used:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{6.1}$$

where n is the batch size, y_i is the ith model prediction and $\hat{y}_i$ is the ith actual value. RMSE is the same loss function as the Mean Squared Error (MSE) [38] loss but has a reduced order, which is ensured by taking the root of the MSE. The RMSE allows direct data correlation with the error as they both have the same order.

A data over-sampling technique was applied to address frequent missing teeth 1, 16, 17, and 32 (Figure 6.1). It ensures that at least one depth map containing the aforementioned teeth is present in each batch. Without the over-sampling, the network would ignore the teeth during evaluation, as it would almost certainly expect the absence of given teeth.

6.2 Overall Results

To find the most accurate system configuration, all three architectures were evaluated in different setups. Firstly, the results of a system that uses a single-view approach were measured. Using this approach, no consensus method is needed. I chose the BatchNorm U-Net with a single-view approach as the baseline method. After the initial evaluation of the baseline method, I noticed that the frequent missing of the 3rd molars causes high radial errors of the corresponding landmarks. To decrease the error, over-sampling of the data containing these teeth was added.

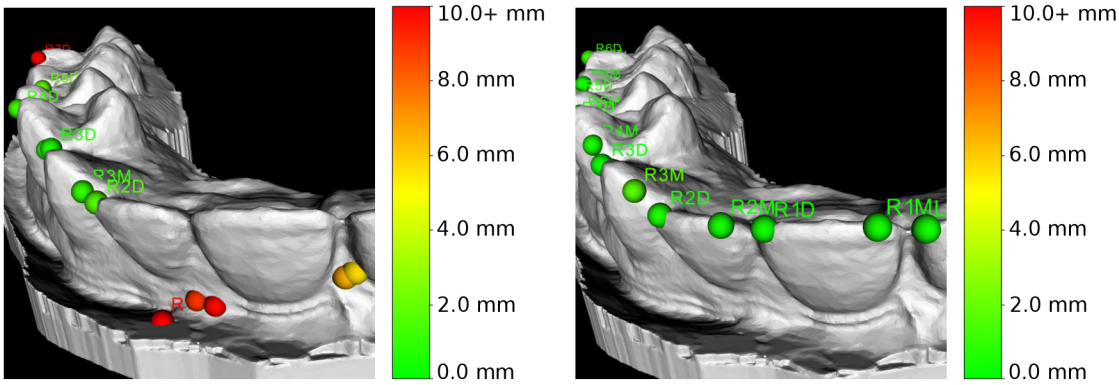
[1] https://pytorch.org/docs/stable/optim.html#torch.optim.lr_scheduler.ReduceLROnPlateau
[2] https://github.com/Bjarten/early-stopping-pytorch. No official implementation is available.
[3] https://pytorch.org/docs/stable/amp.html#torch.cuda.amp.autocast
[4] https://pytorch.org/docs/stable/amp.html#torch.cuda.amp.GradScaler

Architecture & consensus method		Single-view		Multi-view					
				$N = 9$		$N = 25$		$N = 100$	
		$\overline{R}$ (mm)	SD (mm)	$\overline{R}$ (mm)	SD (mm)	$\overline{R}$ (mm)	SD (mm)	$\overline{R}$ (mm)	SD (mm)
BN U-Net	Centroid	2.94	4.62	3.00	3.37	2.74	3.33	2.80	2.96
	RANSAC			2.24	3.86	2.02	3.75	1.61	4.28
Att U-Net	Centroid	2.47	4.06	2.45	2.60	2.39	2.43	2.37	2.42
	RANSAC			1.80	2.80	1.69	2.42	**1.20**	**1.91**
Nes U-Net	Centroid	2.84	4.16	3.09	3.00	2.82	2.74	2.88	2.75
	RANSAC			1.99	3.05	1.72	2.73	1.39	2.23

Table 6.1: Overall results of the individual networks with different multi-view settings. Table compares system performance with different combinations of architectures, consensus methods, and numbers of viewpoints. A combination of Attention U-Net architecture, RANSAC consensus method, and 100 rendered views achieves the best performance. $\overline{R}$ stands for the mean radial error, and SD stands for standard deviation. All values are measured on networks with applied over-sampling of minority data.

Afterward, I started experimenting with the multi-view approach and with other network architectures. I proposed three different numbers of viewpoints – 9, 25, and 100. I expected an inversely proportional trend – the higher the number of viewpoints, the lower the radial errors and standard deviations. I also expected the RANSAC consensus method to achieve better results than Centroid as the networks produce outliers. Table 6.1 summarizes the system performance. It shows that with the **Attention U-Net**, the **RANSAC** consensus method, and **100 viewpoints**, the best results are acquired.

I also measured the SDRs for different acceptance values. The acceptable distance is set to be **2 millimeters**, i.e., if the vast majority of predictions have the radial error less than 2 mm, the system could be eventually used as a part of the clinical application. The SDRs for 2.5 mm and 4 mm are measured as well. I measure them to observe whether the predictions with a radial error higher than the acceptable distance (2 mm) are "close" to 2 mm, or their value is completely unacceptable.



(a) 1 view **(b)** 100 views

Figure 6.2: Comparison of the landmark placement with single view and 100 views. These results are obtained using the Attention U-Net and geometric consensus method. Color of landmark represents the value of its radial error.

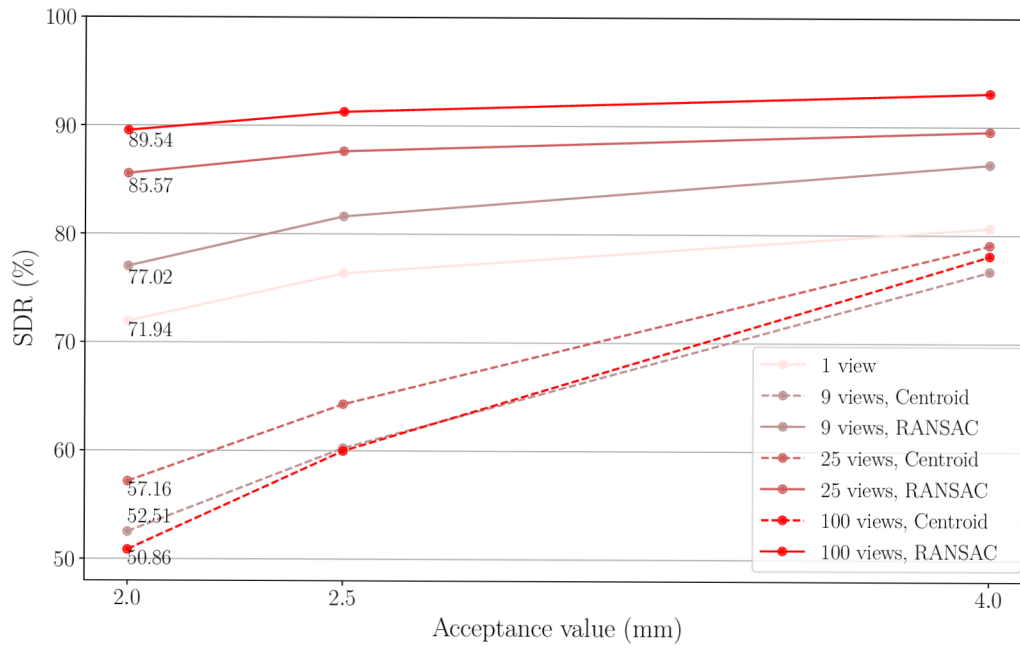


Figure 6.3: SDRs for BatchNorm U-Net. The best performing configuration with this architecture is the RANSAC as a consensus method with 100 views. This configuration achieved an SDR of **89.54% for 2 mm** and 93.07% for 4 mm.

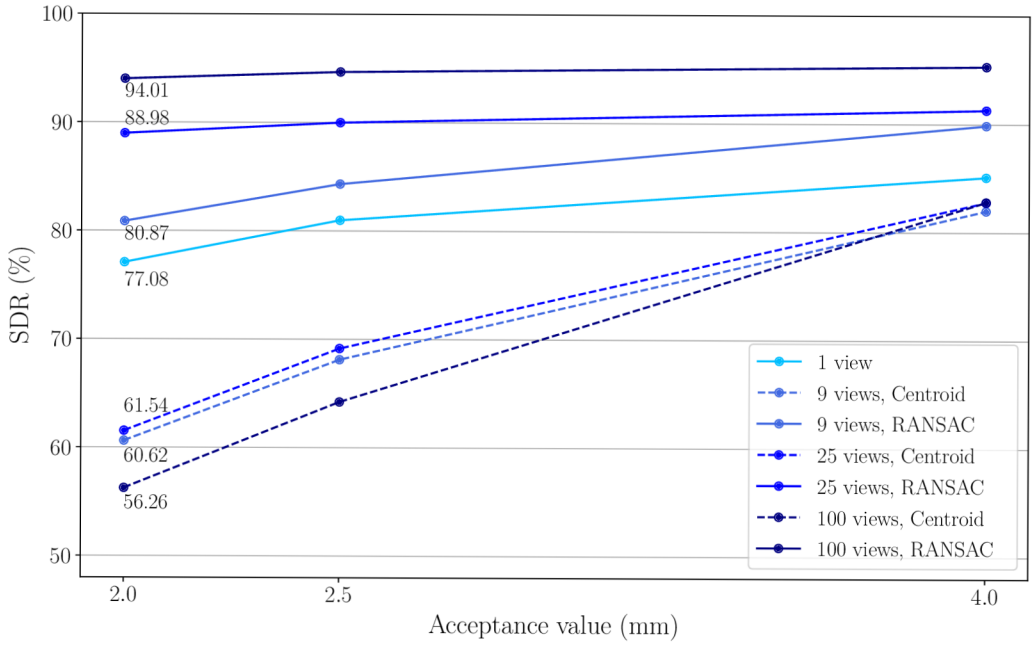


Figure 6.4: SDRs for Attention U-Net. The best performance (**94.01% for 2 mm**) was again achieved with RANSAC applied on 100 predictions. SDR for acceptance value of 4 mm is 95.31%.

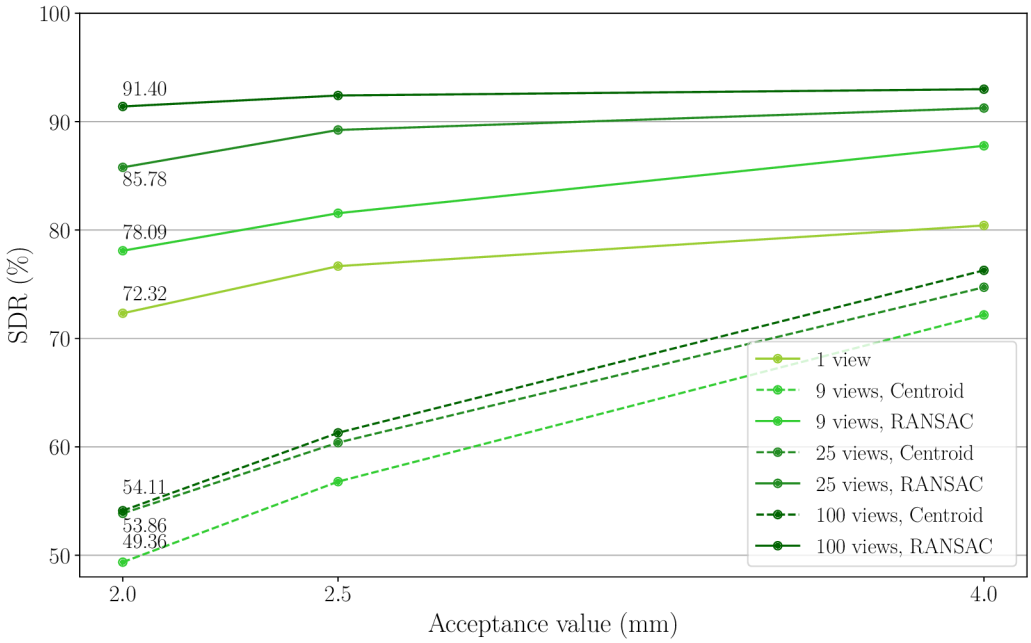


Figure 6.5: SDRs for Nested U-Net. 100 viewpoints with RANSAC achieved the best results – **91.40% for acceptance value of 2 mm**. This configuration achieved 92.99% SDR for 4 mm.

The SDR values for BatchNorm U-Net, Attention U-Net, and Nested U-Net are shown in Figures 6.3, 6.4, and 6.5, respectively. These figures confirm that Attention U-Net with RANSAC and 100 viewpoints are the best performing combination.

It is also evident that **RANSAC** as a consensus method **outperforms Centroid by a large margin**. RANSAC achieves approximately 20–40% better SDR with the acceptance value of 2 mm. It also achieves a descent of $\overline{R}$, on average by approximately 1 mm, compared to Centroid. The difference in results might be explained by the strength of the geometric method in terms of outlier detection, whereas the Centroid method is calculated from the predictions of all views. Figure 6.10 shows example outputs using both of the consenus methods.

Considering all of these, **I recommend the combination of the Attention U-Net and RANSAC consensus method**.

As for the number of viewpoints needed for the multi-view approach, I hypothesized that the increase in the number of views brings better results. This hypothesis is confirmed – it really comes with an increase in the accuracy. However, as for the Centroid consensus method, the SDR values of configurations that use 100 viewpoints achieve the worst results. The observed decrease in the SDR for such configurations is certainly due to the increase of outliers that comes with the increase of viewpoints. In the case of this task, the 100 views approach is considered as the best performing, although it must be combined with the RANSAC consensus method. Figure 6.2 shows example outputs of single view and 100 views combined with the RANSAC consensus method.

6.2.1 Computational Time

The high number of viewpoints, however, comes with a trade-off. For each of the views, the network must produce predictions for a new depth map. It means that it is necessary to count with increased computational time. Additionally, the multi-view evaluation consumes some computational time during the consensus estimation in contrast to the single-view approach. The best performing combination – the Attention U-Net, 100 viewpoints, and RANSAC takes about 4.5 seconds to evaluate. Using fewer camera views, a decrease of computational time is ensured while achieving good results (for example, with Attention U-Net, RANSAC, and 25 views, the increase in $\overline{R}$ is 0.49 mm and the time needed for such evaluation is approximately 1 second). The RANSAC consensus method consumes around 1 second to produce the point estimate from multiple views. Table 6.2 shows the computational times[5] for the Attention U-Net for different number of views.

	Multi-view number of viewpoints		
	$N = 9$	$N = 25$	$N = 100$
Evaluation	380.44 ms	1 074.63 ms	4 446.02 ms
RANSAC	1 178.59 ms	1 252.57 ms	1 306.31 ms
Total	1 559.03 ms	2 327.20 ms	5 752.33 ms

Table 6.2: Average computational time of the evaluation and consensus method outcome calculation of one polygonal mesh with the best-performing network: the Attention U-Net. These values were measured on the evaluation dataset – they represent the average values measured during the evaluation of 61 testing meshes.

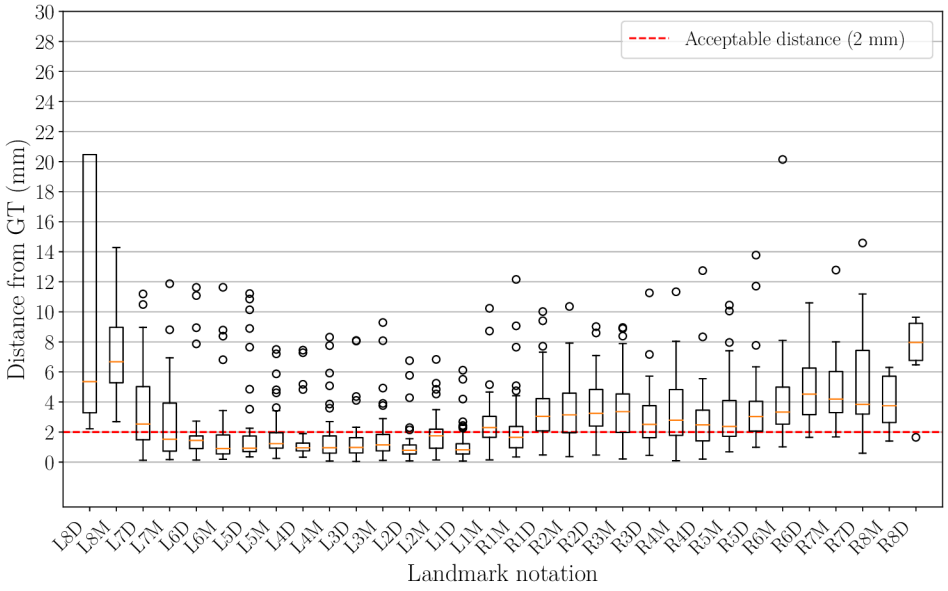
6.3 Analysis of Individual Landmark Accuracies

Networks are trained to predict 32 landmarks on 16 teeth. These teeth, however, are not always present on the evaluated mesh. The third molars are a good example as the presence of polygon meshes that contain these teeth is around 10% (see Figure 6.1). Concerning the fact that the models were obtained by scanning human dentition, such imbalance is natural. Although I applied over-sampling of inferiorly present data, such imbalance on such a small dataset produces some deviation in terms of system performance.

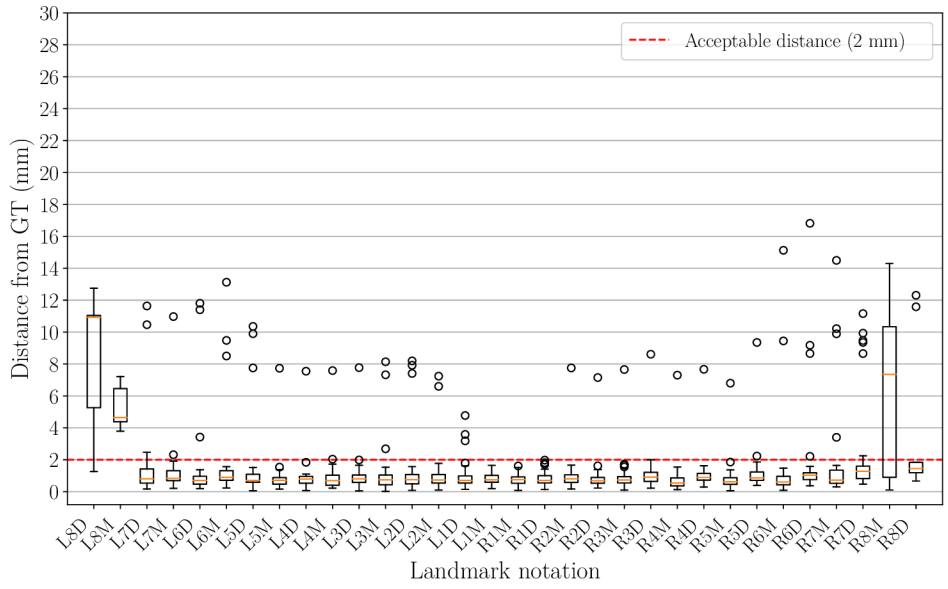
To observe whether the imbalance affects the overall results, prediction accuracies of individual landmarks were measured. As Figure 6.6 illustrates, the values of the radial errors of third molar landmarks (landmarks L8D, L8M, R8M, and R8D) are higher, compared to the rest of the landmarks, even with applied over-sampling. This means that the high absence of depth maps that contain such landmarks influences the radial error of corresponding landmarks during evaluation. In essence, **the results of the third molar landmarks deviate** from the radial errors of the rest of the landmarks. See Figure 6.9 for example outputs. If the presence of polygonal meshes with dentition that contains the third molars in the dataset was higher, the overall performance would be increased. Figures 6.7 and 6.8 illustrate the matching acceptances graphs of two system configurations. In the latter case, it is a system that achieves the best results. Most of the landmarks

[5]Measured on a laptop with Intel Core i7-8750H CPU @ 2.20 GHz and NVIDIA GeForce RTX 2060 with 6 GB of memory

achieve SDRs between 90 to 100% at the acceptance value of 2 mm. Again, the third molar landmarks deviate from this standard and achieve such SDRs at higher acceptance values.



(a)



(b)

Figure 6.6: Box plots of the radial error values for individual landmarks with two different example setups. Graph **(a)** shows the box plots for the following combination – BatchNorm U-Net, 100 views, and Centroid. Graph **(b)** shows, on the other hand, the best combination – Attention U-Net, 100 viewpoints, and RANSAC. Notice that in both cases, the radial errors of landmarks on third molars (L8D, L8M, R8M, R8D) are higher than the rest of the landmarks.

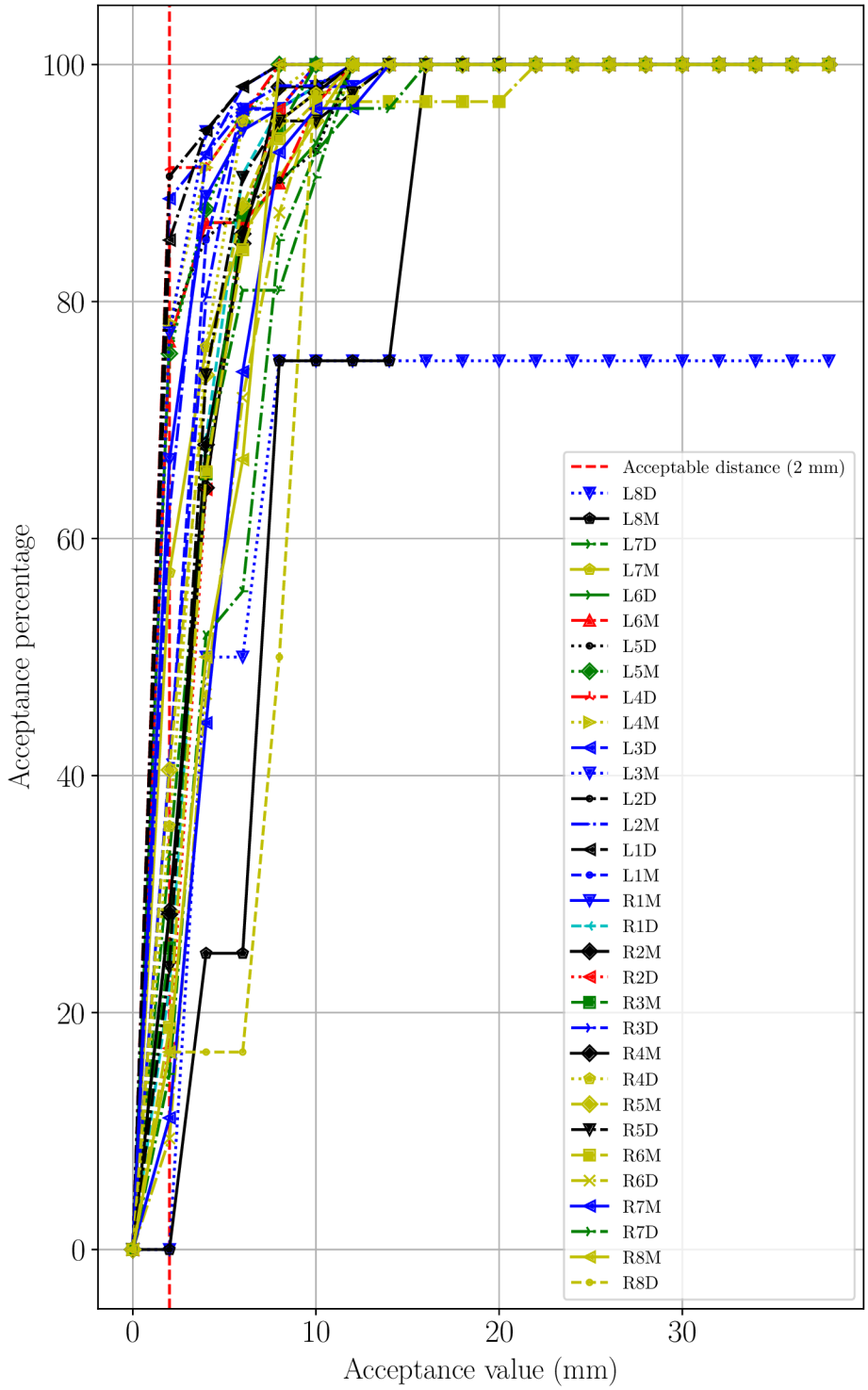


Figure 6.7: Matching acceptances graph for BatchNorm U-Net, 100 viewpoints and Centroid consensus method. Individual landmarks achieve an SDR of 100% around 10 mm. For instance, landmark L8D was never predicted with a radial error of less than 2 mm.

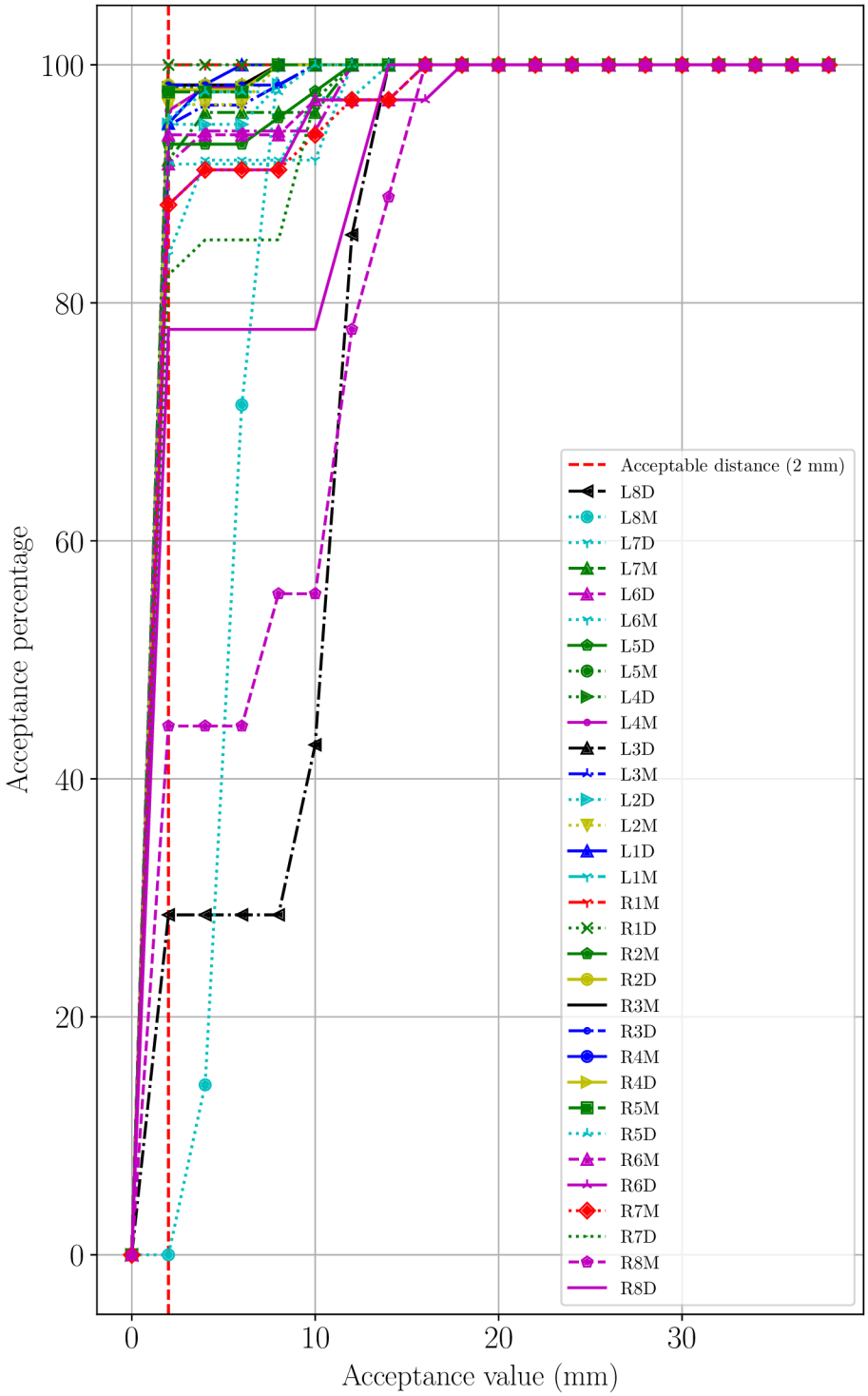
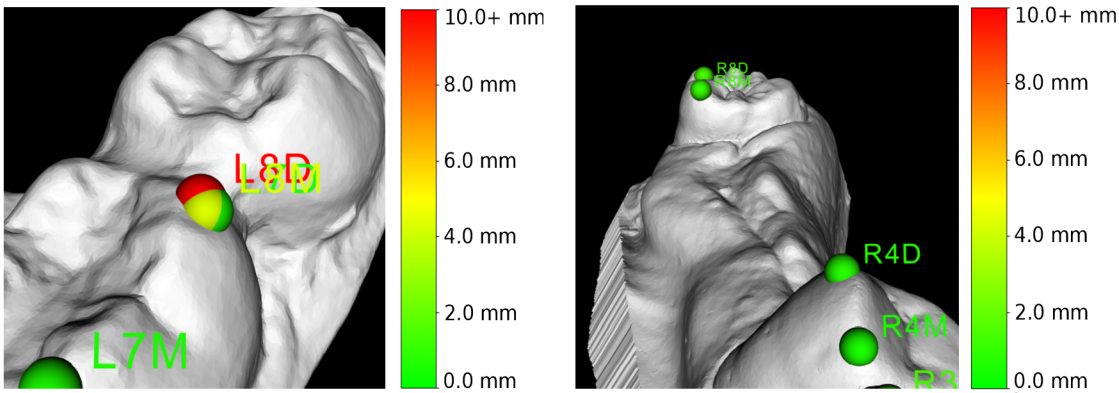
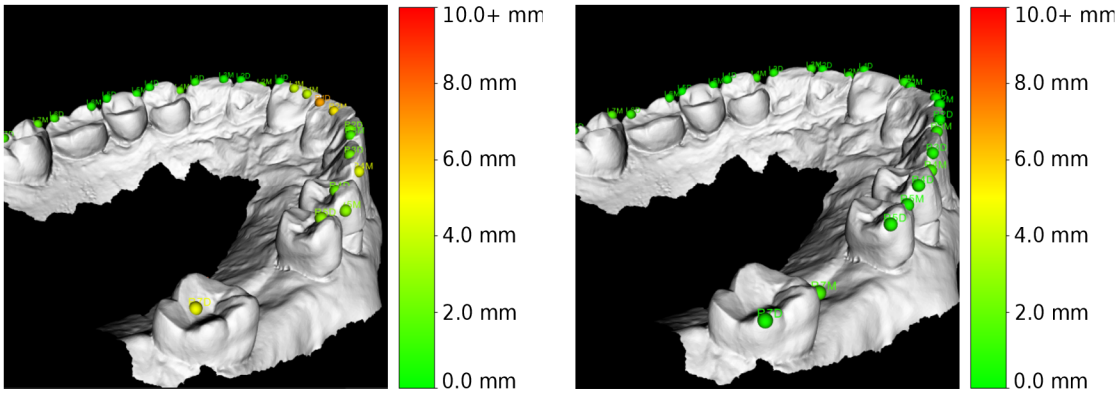


Figure 6.8: Matching acceptances graph for Attention U-Net, 100 viewpoints and RANSAC consensus method. Individual landmarks mostly achieve the SDR between 90% to 100% for the acceptance value of 2 mm. Landmarks on third molars (L8D, L8M, R8M, and R8D) achieve worse results. This corresponds with the radial errors presented in Figure 6.6.



(a) Invalid prediction **(b)** Valid prediction

Figure 6.9: Example of predictions of landmarks on third molars. Image **(b)** illustrates an example of incorrect predictions – the landmarks are incorrectly placed on the second molars. This phenomenon is present almost exclusively on the 3rd molar landmarks. Image **(a)** shows a valid prediction on the third right molar. Predictions are acquired using Attention U-Net, RANSAC, and 100 viewpoints.



(a) Centroid **(b)** RANSAC

Figure 6.10: Example of detections with different consensus methods. On average, the geometric method based on the RANSAC algorithm achieves lower values of radial errors. Detected landmarks were obtained using the Attention U-Net and 100 views, in both images.

6.4 Detection of Landmarks Presence

Besides the accurate landmark placement, this task has an additional challenge. As already discussed in previous sections of this work, the absence of teeth is quite frequent. It is relatively rare to find a full secondary dentition within the dataset. For each detected landmark, it is necessary to decide whether the corresponding tooth is present on the evaluated dentition or not. This decision is a binary classification task as there are two

possible outcomes – landmark is either placed on the surface of the polygon mesh or not. This goes to show that the prediction can be classified as:

- **True positive** (TP): CNN predicts that the landmark is present and the ground truth corresponds to that decision,
- **False positive** (FP): CNN predicts landmark placement, but the appropriate tooth is absent,
- **False negative** (FN): CNN predicts that the landmark should not be placed on the surface of the target polygonal model, but it is a misclassification,
- **True negative** (TN): CNN correctly predicts that the landmark should not be placed on the evaluated polygonal mesh.

The classifier should produce as many TP and TN predictions as possible while suppressing FP and FN predictions.

I wanted to test whether the outputs of neural networks used for the landmark localization task can be used as the aforementioned classifier without any additional intervention. Even though the problem is formulated as a regression (see Section 4.1), the output heatmaps might contain sufficient information for the landmark presence detection. Drevický hypothesized in his work [10], that the peaks of predicted heatmaps may indicate the model's uncertainty in its prediction, with higher values indicating higher certainty.

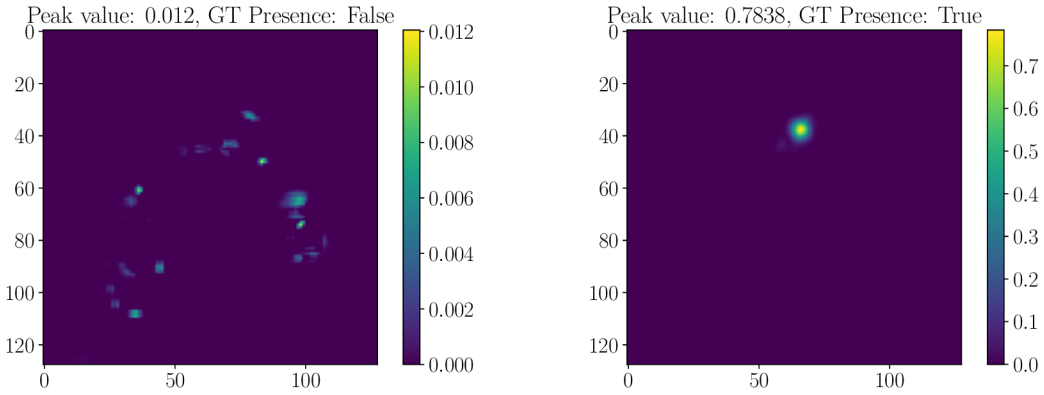


Figure 6.11: Examples of predicted heatmaps. The left picture illustrates an example of a prediction with an extremely low peak value (0.012). Referencing to corresponding ground truth, this landmark is not present on the surface of the polygonal model. The right picture, on the other hand, shows the opposite situation. According to the ground truth, the peak value is relatively high, and this landmark is really present on evaluated polygon mesh. Note that the maximal possible value in a heatmap is 1.

Networks were trained by regressing heatmaps containing a Gaussian activation **with the amplitude of 1**. The predictions should follow a similar trend. If a landmark was missing on the polygonal model during training, there was no Gaussian in the ground truth image. This implies that the predictions should be either heatmaps with a peak value close to 1 or heatmaps with all values close to 0. Figure 6.11 shows an example for both of these situations. It is necessary to find the peak (threshold) value that ensures the best performing binary classification.

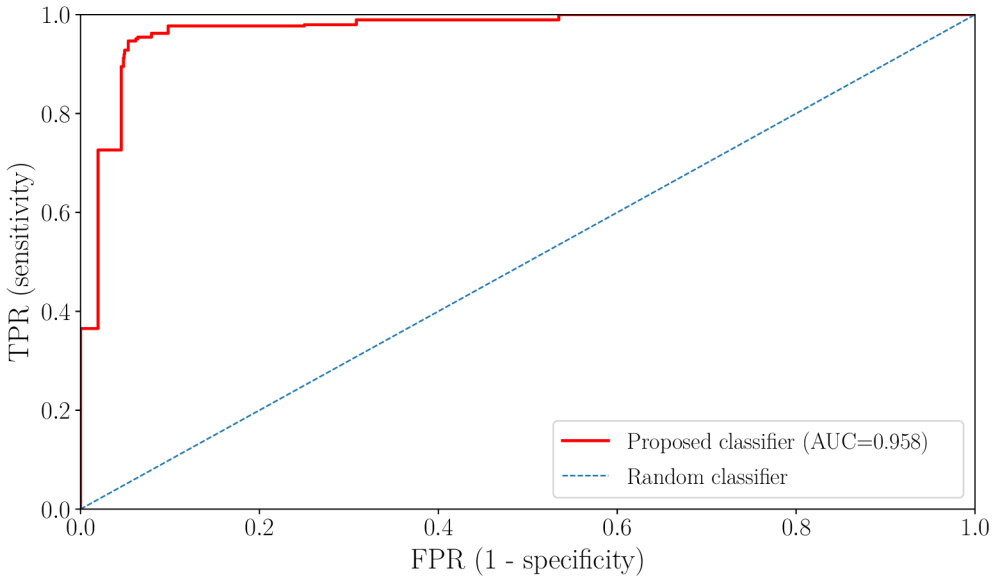


Figure 6.12: ROC curve for proposed classification method. Classification purely from the predicted heatmaps achieves an AUC value of **0.958**. The best threshold value is **0.375**. These values were measured on Attention U-Net with the RANSAC consensus method and 100 viewpoints.

6.4.1 Evaluation of the Landmark Presence Detection

The performance of the landmark absence detection was measured on the Attention U-Net with the RANSAC consensus method and 100 views. To find a threshold that classifies the landmark presence with the best accuracy, the ROC curve was created. I tested 20 different threshold values to find the best trade-off between sensitivity and specificity.

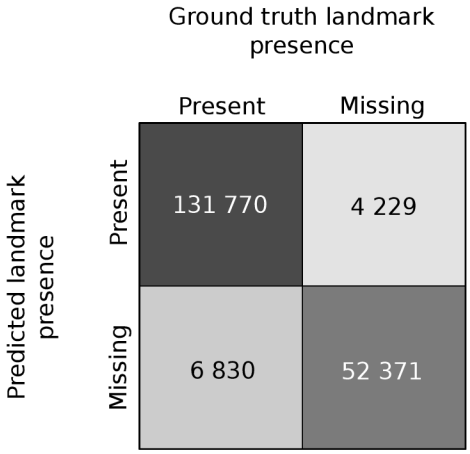


Figure 6.13: Confusion matrix of proposed classifier with threshold set to 0.375. The matrix shows the TP, TN, FP, and FN values of the classification according to the heatmap peak value. These values were measured on the evaluation dataset (61 polygonal models) with 100 viewpoints. The system predicts 32 landmarks. Thus, the total number of values is $61 \times 100 \times 32 = 195\,200$.

Figure 6.12 shows the ROC curve for the classifier. By classifying purely from the heatmap predictions, **AUC value of 0.958** is achieved. See Equation A.11 for the formula. **Threshold value** that brings off the best sensitivity and specificity values **is 0.375**. Note that for a different dataset, I recommend verifying the fit of this value. The corresponding confusion matrix is shown in Figure 6.13.

I calculated the *accuracy* from confusion matrix values. With the best performing threshold value (0.375), the *accuracy* calculated by Equation A.5 is 0.9433. It means that purely from the analysis of peak values in predicted heatmaps, the proposed method can correctly detect landmark presence with the accuracy of **94.33%**.

6.5 Summary

I proposed three neural network architectures, all of which are based on the U-Net network. The first one is the BatchNorm U-Net. This network contains additional batch normalization layers between the convolutional and activation layers. The second proposed network is the Attention U-Net, which integrates attention gates into the original U-Net. The last network is the Nested U-Net, which applies dense skip pathways into the original U-Net.

I started conducting experiments with the point estimates propagated in the $\mathbb{R}^3$ world coordinate system. With a single view, the Attention U-Net achieved the best results with the radial error value of 2.47 ± 4.06 mm and with the success detection rate for 2 mm of 77.08%. The BatchNorm U-Net performed the worst.

Experiments have demonstrated that this approach does not perform well enough to be integrated into a clinical application. I decided to experiment with a multi-view CNN approach, which renders the evaluated model from several views and eventually combines the predictions by a consensus method.

I proposed experiments with two consensus methods – Centroid and RANSAC, and experimented with several views – 9, 25, and 100. As a result of experimenting with all possible combinations of architectures, consensus methods, and numbers of views, I concluded that the best performing combination uses Attention U-Net, RANSAC as a consensus method, and 100 viewpoints. This combination achieved the error value of 1.20 ± 1.81 mm, and it can predict 94.01% of landmarks with the radial error of less than 2 mm. Another aspect that emerged from the analysis is the relevancy of geometric consensus method. It outperformed the statistic approach by a large margin and I highly recommend its usage when solving similar tasks.

Besides the accurate landmark placement, the method should be able to decide whether a given landmark should indeed be placed on the model's surface or not. I wanted to test whether the method can produce such decisions without the need to implement an additional binary classifier. The decision is made purely from the peak values of the predicted heatmaps. With the optimal threshold value, the proposed method can correctly detect landmark presence with the accuracy of 94.33%.

6.6 Future Work

Although the experiments demonstrate that the overall results satisfy the needs of the end clinical application, there is still a place for future development. Results of this work might be improved and extended in both landmarks position detection and the detection of landmarks presence on the 3D model surface.

As for the landmark placement, a method that does not require the initial model rotation (occlusal and incisal surfaces of the teeth must face the camera) would be convenient. This would diminish the time needed for the evaluation, and it would eliminate the human intervention required for the landmarking process. Additionally, the re-balancing strategy used to address the problems caused by minority classes might be replaced by more advanced techniques, such as class-balanced loss based on an effective number of samples [7]. Extending the dataset — primarily by the dentition with third molars — appears to be helpful as well.

There is also a place for the improvement of the accuracy of the landmark presence classifier. One of the improvements might stem from the fact that a pair of landmarks is detected on the surface of each tooth. If a tooth is missing, both landmarks should be classified as absent. Then, the final decision of landmark presence might take into account the information from the pair landmark as well. Another improvement related to the classifier might advantage from the multi-view approach. Besides the point estimate, the RANSAC consensus method outputs the information about inliers and outliers. This information might be valuable for the decision process.

Chapter 7

Conclusion

This Bachelor's thesis aimed to estimate the orthodontics landmarks on the surfaces of polygonal models from a limited medical dataset. A method based on convolutional neural networks was designed for this purpose.

The proposed method transfers the training and evaluation process into the Euclidean 2D space. Instead of directly regressing the landmark coordinates, the proposed networks are trained to regress heatmaps containing Gaussians centered at landmark positions.

Suggested approach uses three architecture designs: the BatchNorm U-Net, the Attention U-Net, and the Nested U-Net. 3D Scans of dentition are evaluated in a multi-view CNN matter, so the model is observed from multiple viewpoints, which produces corresponding number of predictions for each landmark. These predictions are evaluated in a consensus method, which produces the final estimate in 3D space. A consensus method based on the RANSAC algorithm and least-squares fit produces the best results.

A combination of Attention U-Net, RANSAC consensus method, and 100 views performed the best and achieved the error value of 1.20 ± 1.81 mm, and it can predict 94.01% of landmarks with the radial error of less than 2 mm. According to the competent professionals from TESCAN 3DIM, s.r.o., these results are satisfactory for a practical deployment in an existing orthodontics planning software.

To summarize, I proposed a method that can predict orthodontics landmarks on polygonal models with acceptable error values. I compared the performance of different network architectures, numbers of viewpoints, and consensus methods. Taken together, these findings suggest that the multi-view approach combined with the RANSAC consensus method, and architectures based on U-Net can provide acceptable accuracies even on a limited medical dataset. A further collaboration involving method enhancement and integration into existing orthodontics software is discussed with the thesis supervisor.

Bibliography

[1] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M. and Asari, V. K. Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation. *CoRR*. 2018, abs/1802.06955. Available at: http://arxiv.org/abs/1802.06955.

[2] Bateriwala, M. and Bourgeat, P. Enforcing temporal consistency in Deep Learning segmentation of brain MR images. *CoRR*. 2019, abs/1906.07160. Available at: http://arxiv.org/abs/1906.07160.

[3] Bier, B., Unberath, M., Zaech, J., Fotouhi, J., Armand, M. et al. X-ray-transform Invariant Anatomical Landmark Detection for Pelvic Trauma Surgery. In: Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola López, C. and Fichtinger, G., ed. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part IV*. Cham: Springer, 2018, vol. 11073, p. 55–63. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-00937-3_7. ISBN 978-3-030-00937-3. Available at: https://doi.org/10.1007/978-3-030-00937-3_7.

[4] Blalock, D. W., Ortiz, J. J. G., Frankle, J. and Guttag, J. V. What is the State of Neural Network Pruning? In: Dhillon, I. S., Papailiopoulos, D. S. and Sze, V., ed. *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. Mlsys.org, 2020, vol. 2, p. 129–146. Available at: https://proceedings.mlsys.org/book/296.pdf.

[5] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. and Vandergheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*. Institute of Electrical and Electronics Engineers (IEEE). July 2017, vol. 34, no. 4, p. 18–42. DOI: 10.1109/msp.2017.2693418. ISSN 1558-0792. Available at: http://dx.doi.org/10.1109/MSP.2017.2693418.

[6] Burke, D. *Malocclusion of the Teeth* [online]. 2016 [cit. 2021-01-24]. Available at: https://www.healthline.com/health/malocclusion-of-teeth.

[7] Cui, Y., Jia, M., Lin, T.-Y., Song, Y. and Belongie, S. J. Class-Balanced Loss Based on Effective Number of Samples. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, p. 9268–9277. DOI: 10.1109/CVPR.2019.00949. ISBN 978-1-7281-3293-8. Available at: http://openaccess.thecvf.com/content_CVPR_2019/html/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR_2019_paper.html.

[8] Dabbah, M. A., Murphy, S., Pello, H., Courbon, R., Beveridge, E. et al. Detection and location of 127 anatomical landmarks in diverse CT datasets. In: Ourselin, S. and Styner, M. A., ed. *Medical Imaging 2014: Image Processing*. SPIE, 2014, vol. 9034, p. 284 – 294. DOI: 10.1117/12.2039157. Available at: https://doi.org/10.1117/12.2039157.

[9] Donner, R., Menze, B. H., Bischof, H. and Langs, G. Global localization of 3D anatomical structures by pre-filtered Hough Forests and discrete optimization. *Medical Image Analysis*. 2013, vol. 17, no. 8, p. 1304–1314. DOI: https://doi.org/10.1016/j.media.2013.02.004. ISSN 1361-8415. Available at: https://www.sciencedirect.com/science/article/pii/S1361841513000182.

[10] Drevický, D. *Deep Learning Model Uncertainty in Medical Image Analysis*. Brno, CZ, 2019. Master's thesis. Brno University of Technology, Faculty of Information Technology. Available at: https://www.fit.vut.cz/study/thesis/22094/.

[11] Fischler, M. A. and Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*. 1st ed. New York, NY, USA: Association for Computing Machinery. June 1981, vol. 24, no. 6, p. 381–395. DOI: 10.1145/358669.358692. ISSN 0001-0782. Available at: https://doi.org/10.1145/358669.358692.

[12] Gao, Y. and Shen, D. Collaborative regression-based anatomical landmark detection. *Physics in Medicine & Biology*. IOP Publishing. 2015, vol. 60, no. 24, p. 9377. DOI: 10.1088/0031-9155/60/24/9377. Available at: https://iopscience.iop.org/article/10.1088/0031-9155/60/24/9377.

[13] Graham, B. Sparse 3D convolutional neural networks. In: Xie, X., Jones, M. W. and Tam, G. K. L., ed. *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, September 2015, p. 150.1–150.9. DOI: 10.5244/C.29.150. ISBN 1901725537. Available at: https://dx.doi.org/10.5244/C.29.150.

[14] Hajati, A. *Digital orthodontics* [online]. 2020 [cit. 2021-01-24]. Available at: https://me.dental-tribune.com/clinical/digital-orthodontics/.

[15] Han, D., Gao, Y., Wu, G., Yap, P.-T. and Shen, D. Robust Anatomical Landmark Detection for MR Brain Image Registration. In: Golland, P., Hata, N., Barillot, C., Hornegger, J. and Howe, R. D., ed. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2014 - 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part I*. Springer, 2014, vol. 8673, p. 186–193. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-10404-1_24. ISBN 978-3-319-10404-1. Available at: https://doi.org/10.1007/978-3-319-10404-1_24.

[16] Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*. 2nd ed. USA: Cambridge University Press, 2004. ISBN 0521540518.

[17] Hossin, M. and Sulaiman, M. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*. Academy & Industry Research Collaboration Center (AIRCC). March 2015, vol. 5, no. 2, p. 1–11. DOI: 10.5121/ijdkp.2015.5201. Available at: https://www.aircconline.com/ijdkp/V5N2/5215ijdkp01.pdf.

[18] Karavides, T., Leung, K. Y. E., Paclík, P., Hendriks, E. A. and Bosch, J. G. Database guided detection of anatomical landmark points in 3D images of the heart. In: *Proceedings of the 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Rotterdam, The Netherlands, 14-17 April, 2010*. IEEE, 2010, p. 1089–1092. DOI: 10.1109/ISBI.2010.5490182. ISBN 978-1-4244-4125-9. Available at: https://doi.org/10.1109/ISBI.2010.5490182.

[19] Khanna, S. Artificial intelligence: contemporary applications and future compass. *International dental journal*. Wiley Online Library. 2010, vol. 60, no. 4, p. 269–272. ISSN 0020-6539. Available at: https://www.sciencedirect.com/science/article/pii/S0020653920341861.

[20] Lee, C.-Y., Xie, S., Gallagher, P. W., Zhang, Z. and Tu, Z. Deeply-Supervised Nets. In: Lebanon, G. and Vishwanathan, S. V. N., ed. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*. JMLR.org, 2015, vol. 38. JMLR Workshop and Conference Proceedings. Available at: http://proceedings.mlr.press/v38/lee15a.html.

[21] Lee, H., Park, M. and Kim, J. Cephalometric landmark detection in dental x-ray images using convolutional neural networks. In: III, S. G. A. and Petrick, N. A., ed. *Medical Imaging 2017: Computer-Aided Diagnosis, Orlando, Florida, United States, 11-16 February 2017*. SPIE, 2017, vol. 10134, p. 101341W. SPIE Proceedings. DOI: 10.1117/12.2255870. Available at: https://doi.org/10.1117/12.2255870.

[22] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F. et al. A survey on deep learning in medical image analysis. *Medical Image Anal.* 2017, vol. 42, p. 60–88. DOI: 10.1016/j.media.2017.07.005. Available at: https://doi.org/10.1016/j.media.2017.07.005.

[23] Lu, X. and Jolly, M. Discriminative Context Modeling Using Auxiliary Markers for LV Landmark Detection from a Single MR Image. In: Camara, O., Mansi, T., Pop, M., Rhode, K. S., Sermesant, M. et al., ed. *Statistical Atlases and Computational Models of the Heart. Imaging and Modelling Challenges - Third International Workshop, STACOM 2012, Held in Conjunction with MICCAI 2012, Nice, France, October 5, 2012, Revised Selected Papers*. Berlin, Heidelberg: Springer, 2012, vol. 7746, p. 105–114. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-36961-2_13. ISBN 978-3-642-36961-2. Available at: https://doi.org/10.1007/978-3-642-36961-2_13.

[24] Lv, J., Shao, X., Xing, J., Cheng, C. and Zhou, X. A Deep Regression Architecture with Two-Stage Re-initialization for High Performance Facial Landmark Detection. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, p. 3691–3700. DOI: 10.1109/CVPR.2017.393. ISBN 978-1-5386-0457-1. Available at: https://doi.org/10.1109/CVPR.2017.393.

[25] Mahapatra, D. Landmark Detection in Cardiac MRI Using Learned Local Image Statistics. In: Camara, O., Mansi, T., Pop, M., Rhode, K. S., Sermesant, M. et al., ed. *Statistical Atlases and Computational Models of the Heart. Imaging and Modelling Challenges - Third International Workshop, STACOM 2012, Held in*

Conjunction with MICCAI 2012, Nice, France, October 5, 2012, Revised Selected Papers. Berlin, Heidelberg: Springer, 2012, vol. 7746, p. 115–124. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-36961-2_14. ISBN 978-3-642-36961-2. Available at: https://doi.org/10.1007/978-3-642-36961-2_14.

[26] Nair, V. and Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In: Fürnkranz, J. and Joachims, T., ed. *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*. Madison, WI, USA: Omnipress, 2010, p. 807–814. ICML'10. ISBN 978-1-605-58907-7. Available at: https://icml.cc/Conferences/2010/papers/432.pdf.

[27] Noothout, J. M. H., Vos, B. D. de, Wolterink, J. M., Leiner, T. and Isgum, I. CNN-based Landmark Detection in Cardiac CTA Scans. *CoRR*. 2018, abs/1804.04963. Available at: http://arxiv.org/abs/1804.04963.

[28] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P. et al. Attention U-Net: Learning Where to Look for the Pancreas. *CoRR*. 2018, abs/1804.03999. Available at: http://arxiv.org/abs/1804.03999.

[29] Paulsen, R. R., Juhl, K. A., Haspang, T. M., Hansen, T. F., Ganz, M. et al. Multi-view Consensus CNN for 3D Facial Landmark Placement. In: Jawahar, C. V., Li, H., Mori, G. and Schindler, K., ed. *Computer Vision - ACCV 2018 - 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part I*. Cham: Springer, 2018, vol. 11361, p. 706–719. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-20887-5_44. ISBN 978-3-030-20887-5. Available at: https://doi.org/10.1007/978-3-030-20887-5_44.

[30] Payer, C., Stern, D., Bischof, H. and Urschler, M. Regressing Heatmaps for Multiple Landmark Localization Using CNNs. In: Ourselin, S., Joskowicz, L., Sabuncu, M. R., Ünal, G. B. and Wells, W., ed. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016 - 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II*. Springer, 2016, vol. 9901, p. 230–238. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-46723-8_27. ISBN 978-3-319-46723-8. Available at: https://doi.org/10.1007/978-3-319-46723-8_27.

[31] Pfister, T., Charles, J. and Zisserman, A. Flowing ConvNets for Human Pose Estimation in Videos. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, p. 1913–1921. DOI: 10.1109/ICCV.2015.222. ISBN 978-1-4673-8391-2. Available at: https://doi.org/10.1109/ICCV.2015.222.

[32] Rheude, B., Lionel Sadowsky, P., Ferriera, A. and Jacobson, A. An evaluation of the use of digital study models in orthodontic diagnosis and treatment planning. *The Angle Orthodontist*. 2005, vol. 75, no. 3, p. 300–304. DOI: 10.1043/0003-3219(2005)75[300:AEOTUO]2.0.CO;2. Available at: https://meridian.allenpress.com/angle-orthodontist/article/75/3/300/58215/An-Evaluation-of-the-Use-of-Digital-Study-Models.

[33] Rickne C. Scheid, G. W. D. *Woelfel's Dental Anatomy: Its Relevance to Dentistry*. 8th ed. Lippincott Williams & Wilkins, 2011. ISBN 978-1-608-31746-2.

[34] Rolland, S. L., Treasure, E., Burden, D. J., Fuller, E. and Vernazza, C. R. The orthodontic condition of children in England, Wales and Northern Ireland 2013. *British Dental Journal*. October 2016, vol. 221, no. 7, p. 415–419. DOI: 10.1038/sj.bdj.2016.734. ISSN 1476-5373. Available at: https://doi.org/10.1038/sj.bdj.2016.734.

[35] Ronneberger, O., Fischer, P. and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., III, W. M. W. and Frangi, A. F., ed. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. Springer, 2015, vol. 9351, p. 234–241. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-24574-4_28. ISBN 978-3-319-24574-4. Available at: https://doi.org/10.1007/978-3-319-24574-4_28.

[36] Sahi, A. *Universal Numbering System for Teeth* [online]. 2019 [cit. 2021-01-23]. Available at: https://www.news-medical.net/health/Universal-Numbering-System-for-Teeth.aspx.

[37] Saltzman, M. *AutoTrim Aligners*. 2020 [cit. 2021-03-26]. Available at: https://www.youtube.com/watch?v=89EWXNwBHsU.

[38] Sammut, C. and Webb, G. I., ed. Mean Squared Error. In: Sammut, C. and Webb, G. I., ed. *Encyclopedia of Machine Learning*. Boston, MA: Springer, 2010, p. 653. DOI: 10.1007/978-0-387-30164-8_528. ISBN 978-0-387-30164-8. Available at: https://doi.org/10.1007/978-0-387-30164-8_528.

[39] Siddique, N., Sidike, P., Elkin, C. and Devabhaktuni, V. U-Net and its variants for medical image segmentation: theory and applications. *CoRR*. 2020, abs/2011.01118. Available at: https://arxiv.org/abs/2011.01118.

[40] Simon J. Littlewood, L. M. *An Introduction to Orthodontics*. 5th ed. Oxford University Press, 2019. ISBN 978-0-198-84726-7.

[41] Sinha, A., Bai, J. and Ramani, K. Deep Learning 3D Shape Surfaces Using Geometry Images. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., ed. *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*. Springer, 2016, vol. 9910, p. 223–240. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-46466-4_14. ISBN 978-3-319-46466-4. Available at: https://doi.org/10.1007/978-3-319-46466-4_14.

[42] Soltani, A. A., Huang, H., Wu, J., Kulkarni, T. D. and Tenenbaum, J. B. Synthesizing 3D Shapes via Modeling Multi-view Depth Maps and Silhouettes with Deep Generative Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, p. 2511–2519. DOI: 10.1109/CVPR.2017.269. ISBN 978-1-5386-0458-8. Available at: https://doi.org/10.1109/CVPR.2017.269.

[43] Su, H., Maji, S., Kalogerakis, E. and Learned Miller, E. G. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile,*

December 7-13, 2015. IEEE Computer Society, 2015, p. 945–953. DOI: 10.1109/ICCV.2015.114. ISBN 978-1-4673-8391-2. Available at: https://doi.org/10.1109/ICCV.2015.114.

[44] Subramanian, V. *Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch*. 1st ed. Packt Publishing Ltd, 2018. ISBN 1788624335.

[45] Sun, Y., Wang, X. and Tang, X. Deep Convolutional Network Cascade for Facial Point Detection. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*. IEEE Computer Society, 2013, p. 3476–3483. DOI: 10.1109/CVPR.2013.446. ISBN 978-1-5386-5672-3. Available at: https://doi.org/10.1109/CVPR.2013.446.

[46] Taneva, E., Kusnoto, B. and Evans, C. A. 3D Scanning, Imaging, and Printing in Orthodontics. In: *Issues in Contemporary Orthodontics*. InTech, September 2015. DOI: 10.5772/60010. Available at: https://doi.org/10.5772/60010.

[47] Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y. and Tong, X. O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. *ACM Transactions on Graphics*. Association for Computing Machinery (ACM). July 2017, vol. 36, no. 4, p. 1–11. DOI: 10.1145/3072959.3073608. ISSN 1557-7368. Available at: http://dx.doi.org/10.1145/3072959.3073608.

[48] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L. et al. 3D ShapeNets: A deep representation for volumetric shapes. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, p. 1912–1920. DOI: 10.1109/CVPR.2015.7298801. ISBN 978-1-4673-6964-0. Available at: https://doi.org/10.1109/CVPR.2015.7298801.

[49] Zhong, Z., Li, J., Zhang, Z., Jiao, Z. and Gao, X. An Attention-Guided Deep Regression Model for Landmark Detection in Cephalograms. In: Shen, D., Liu, T., Peters, T. M., Staib, L. H., Essert, C. et al., ed. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019 - 22nd International Conference, Shenzhen, China, October 13-17, 2019, Proceedings, Part VI*. Springer, 2019, vol. 11769, p. 540–548. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-32226-7_60. ISBN 978-3-030-32226-7. Available at: https://doi.org/10.1007/978-3-030-32226-7_60.

[50] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N. and Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In: Stoyanov, D., Taylor, Z., Carneiro, G., Syeda-Mahmood, T. F., Martel, A. L. et al., ed. *Deep Learning in Medical Image Analysis - and - Multimodal Learning for Clinical Decision Support - 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings*. Springer, 2018, vol. 11045, p. 3–11. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-00889-5_1. ISBN 978-3-030-00889-5. Available at: https://doi.org/10.1007/978-3-030-00889-5_1.

Appendix A

Evaluation Metrics

This appendix contains metrics used for system performance evaluation. It is divided into two sections. The first Section introduces metrics for the measurement of accuracy of predictions. Then, metrics for a binary classifier are presented. Metrics from the latter case are used in this work to evaluate the landmark presence on the surface of 3D models.

A.1 Landmark Detection Accuracy Metrics

To measure the landmark detection accuracy, several metrics are used – the *radial error*, the *mean radial error*, the *standard deviation*, and *matching acceptances graphs*, which depict the relation between the *success detection rate* and the corresponding acceptance value.
In a Euclidean 3-space $\mathbb{R}^3$, the *radial error* R is defined as

$$R = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2 + (\hat{z} - z)^2} \tag{A.1}$$

where $A(\hat{x},\ \hat{y},\ \hat{z})$ in $\mathbb{R}^3$ is the ground truth landmark position and $B(x,\ y,\ z)$ in $\mathbb{R}^3$ is the predicted value. Two other metrics associated with the radial error are used. The *mean radial error* $\overline{R}$ is calculated as

$$\overline{R} = \frac{\sum_{i=1}^{N} R_i}{N} \tag{A.2}$$

where N is the total number of predictions. The *standard deviation* (SD) is then computed by the following formula:

$$SD = \sqrt{\frac{\sum_{i=1}^{N} (R_i - \overline{R})^2}{N}} \tag{A.3}$$

with N denoting the total number of predictions again. The aforementioned measures are used to evaluate the overall performance of the system.

The *success detection rate* (SDR) is calculated as follows:

$$SDR_v = \frac{\#\{i : \|B(i) - A(i)\| < v\}}{N} \times 100\% \tag{A.4}$$

where B in $\mathbb{R}^3$ is the predicted landmark position, A in $\mathbb{R}^3$ is the ground truth landmark position, v is the corresponding acceptance value and N is the total number of landmarks. It can be used to form *matching acceptances graphs*, which denote the relation between several acceptance values and corresponding SDRs.

A.2 Binary Classifier Metrics

Four values are typically measured and used for the evaluation of a binary classifier. These are:

- True Positive (TP): number of positive instances that are correctly classified,
- True Negative (TN): number of negative instances that are correctly classified,
- False Positive (FP): number of misclassified negative instances,
- False Negative (FN): number of misclassified positive instances.

Aforementioned values (TP, TN, FP, and FN) are typically used in threshold metrics for classification evaluations [17]. A fundamental way of the evaluation of a classifier is *confusion matrix*. It is a 2×2 matrix[1], where each row represents an actual class, while each column represents a predicted class. Items of the matrix are measured TP, TN, FP, and FN values.

Another binary classifier metric is the *accuracy*. In general, the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated and is calculated as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.5}$$

where TP, TN, FP, and FN refer to the values described above. *Error rate* (err) measures the ratio of incorrect predictions over the total number of evaluated instances:

$$err = \frac{FP + FN}{TP + TN + FP + FN} \tag{A.6}$$

To measure the fraction of positive patterns that are correctly classified, the *sensitivity* is measured. It is calculated by following formula:

$$sensitivity = \frac{TP}{TP + FN}. \tag{A.7}$$

Another metric is the $specificity$. This metric is used to measure the fraction of negative patterns that are correctly classified. It is defined as

$$specificity = \frac{TN}{TN + FP}. \tag{A.8}$$

Precision and *recall* are used to measure the performance of a binary classifier as well. Precision is used to measure the positive patterns that are predicted correctly out of all predictions from positive class:

$$precision = \frac{TP}{TP + FP}. \tag{A.9}$$

Recall, on the other hand, measures the fraction of positive patterns that are correctly classified, and is calculated as follows:

$$recall = \frac{TP}{TP + TN}. \tag{A.10}$$

[1] this size is used for a binary classifier and it increases with the number of classes

Note that there are plenty of different metrics by which the classifier performance might be evaluated. A summary of such metrics can be found in [17].

A valuable metric is the *Receiver Operating Characteristics (ROC) curve*, which measures the performance at various thresholds. The ROC curve is plotted with *sensitivity* against $1 - specificity$, where *sensitivity* is on the y-axis and $1 - specificity$ is on the x-axis.

ROC curve comes with additional metric – the *Area Under the ROC Curve (AUC)*. *AUC* represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the *AUC*, the better the model is at distinguishing between landmark presence/absence. AUC can be calculated as below:

$$AUC = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n} \tag{A.11}$$

where S_p is the sum of all positive examples ranked, and n_p and n_n define the number of positive and negative examples, respectively.

Appendix B

Contents of the Included Storage Media

- `data/test/` Folder with polygonal models for testing.[1]
- `data/train/` Folder with data for network training.[2]
- `saved-models/` Folder with trained networks.
- `src/` Folder with source files.
- `src-tech-report/` Folder with LaTeX source files.
- `videos/` Folder with demonstration videos.
- `annotate.sh` Script to run the annotation tool.
- `evaluate.sh` Script to run the evaluation.
- `LICENCE` Project licence.
- `poster.pdf` Poster summarizing this work.
- `README.md` Project description.
- `requirements.txt` List of required Python libraries.
- `tech-report.pdf` Technical report.
- `tech-report-print.pdf` Technical report for two-sided printing.[3]

[1] Note that the provided data are for demonstration purposes only and they are just a small portion of the whole dataset. The complete dataset of 3D scans provided by TESCAN 3DIM, s.r.o. as well as the generated 2D data in this work are not available for privacy reasons.

[2] See footnote 1.

[3] Content is the same as in the version for the electronic submission

Appendix C

Poster