Intraoperative Surgical Tool Pose Estimation Based on Fluoroscopic Landmark Detection

Master Thesis

Qiaowen Wang
19-760-545

Submitted on
25.11.2022

Thesis Supervisor
Prof. Dr. Manuel Günther
Prof. Dr. Philipp Fürnstahl
Dr. Hooman Esfandiari

University of Zurich
Department of Informatics

Master Thesis

Author: Qiaowen Wang, qiaowen.wang@uzh.ch

Project period: 25.05.2022 - 25.11.2022

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

Words cannot express my gratitude to Dr. Hoomam Esfandiari and Prof Dr. Manuel Günther for their unlimited patience, relentless support and valuable suggestions. Especially I'm deeply indebted to Dr. Hooman Esfandiari, he was always willing to rescue me whenever I was trapped by obstacles. He is one of the best supervisors one could ask for, his passion for his work has deeply inspired me, and I'll always be grateful for him taking me as one of his students. I would also like to extend my deepest gratitude to Prof. Dr. Philipp Fürnstahl, none of the endeavours would have been possible without him granting me the opportunity to work under the ROCS group. It's my all-time pleasure to have them as my supervisors.

I would like to thank the University of Zurich for providing me with this chance to study in such a diverse, open-minded, international and bounded environment. The past three years of my master's study at the University of Zurich are a lifelong memory I shall never forget.

Thanks should also go to my friends, Yingying Chen, Calvin Mitchell Ongkingco, Xianxiao Xu, Xiaotian Derrick Yang and Feifei Zheng, for their moral support.

Lastly, I would be remiss in not mentioning my family, especially my parents. Their faith in me and understanding have always been a beacon of light when my spirit was down.

Abstract

Optical surgical instrument tracking systems have been invented and polished for decades, yet due to a variety of reasons, their popularity is still not reaching a satisfactory state. Among all the obstacles, the high monetary expense of introducing such systems to the operating rooms is considered a significant impediment. However, one type of equipment that is indispensable for arming an operating room is the intraoperative imaging system. Therefore, the idea of providing intraoperative navigation based on intraoperative fluoroscopy has been developed and named X23D. This study aims to explore the potential method to support the realization of X23D, especially the feasibility of integrating an advanced neural network into the pipeline. Learning from existing navigation systems, a prototype of a reference frame for locating instruments in fluoroscopic images is sketched. We focus on the potentiality of locating the reference frame using a single fluoroscopic image and 6 landmarks. We performed error stimulation to build our preliminary expectation of the neural network's performance. We defined criteria that can be used to filter the pose of the reference frame in 2D images, which can separate poses into challenging and unchallenging poses. Based on the criteria, we generated the data needed for model training, validation, and testing. The neural network structure that can fulfil the performance expectation is also designed and trained. Even though the accuracy of the proposed approach still craves improvement before it can be deployed into practice, the value of this project as a stepping stone is not to be neglected.

Zusammenfassung

Optische Systeme zur Verfolgung von chirurgischen Instrumenten werden seit Jahrzehnten erfunden und ausgefeilt, doch aus einer Vielzahl von Gründen hat ihre Popularität noch immer keinen zufriedenstellenden Stand erreicht. Unter all den Hindernissen wird der hohe finanzielle Aufwand für die Einführung solcher Systeme in den Operationssälen als ein wesentliches Hindernis angesehen. Eine Ausrüstung, die für die Ausstattung eines Operationssaals unverzichtbar ist, ist jedoch das intraoperative Bildgebungssystem. Daher wurde die Idee einer intraoperativen Navigation auf der Grundlage der intraoperativen Fluoroskopie entwickelt und X23D genannt. Diese Studie zielt darauf ab, die potenzielle Methode zur Unterstützung der Realisierung von X23D zu erforschen, insbesondere die Machbarkeit der Integration eines fortschrittlichen neuronalen Netzwerks in die Pipeline. In Anlehnung an bestehende Navigationssysteme wird ein Prototyp eines Referenzrahmens zur Lokalisierung von Instrumenten in Durchleuchtungsbildern entworfen. Wir konzentrieren uns auf die Möglichkeit, den Referenzrahmen mit Hilfe eines einzigen Durchleuchtungsbildes und 6 Landmarken zu lokalisieren. Wir haben eine Fehlerstimulation durchgeführt, um unsere vorläufige Erwartung an die Leistung des neuronalen Netzes zu ermitteln. Wir haben Kriterien definiert, die zur Filterung der Position des Referenzrahmens in 2D-Bildern verwendet werden können, um die Posen in schwierige und nicht schwierige Posen zu unterteilen. Auf der Grundlage dieser Kriterien haben wir die für das Training, die Validierung und den Test des Modells erforderlichen Daten generiert. Die Struktur des neuronalen Netzes, die die Leistungserwartungen erfüllen kann, wurde ebenfalls entworfen und trainiert. Auch wenn die Genauigkeit des vorgeschlagenen Ansatzes noch verbessert werden muss, bevor er in der Praxis eingesetzt werden kann, ist der Wert dieses Projekts als Sprungbrett nicht zu vernachlässigen.

Contents

Chapter 1

Introduction

A precise execution of the preoperative surgical plan is crucial to minimize possible postoperative complications. Dedicated preoperative surgical planning is the prerequisite of a satisfactory surgery outcome, but the guarantee of accurate delivery in a real-world scenario requires not only accumulated experiences of the surgeons but also aids from peripheral guiding devices.

One commonly used surgical tool in orthopaedics is the Kirschner wire. It is widely used to hold bone fragments together or to provide an anchor for skeletal traction. During the surgeries, the k-wires are placed first by the surgeons over the planned entry points, either by doing small incisions or direct punctures. There are two possible ways to insert the k-wires: one is to perform the insertion by hand, and another involves using a surgical oscillating drill. Depending on the availability of the navigation systems, this process can either be navigated or not. Yet, when complex procedures on the vertebral column require the use of k-wires, any micro deviation in the positions of the k-wires is prone to cause severe consequences. The ability to circumvent any possible damage to main blood vessels and nerves becomes exceptionally necessary to ensure minimal side effects. With the help of intraoperative imaging, in combination with image navigation, direct vision, and tactile feedback, surgeons have the opportunity to increase their confidence in the position and trajectory of k-wires or other surgical tools, which has the potential to improve surgical outcomes.

There are a number of existing sophisticated technologies that provide intraoperative navigation. However, such devices are not commonly affordable by many medical facilities, and none of them is based solely on intraoperative radiographs that are essential to many operating rooms. In addition, they may heavily rely on supplemental monitoring and tracking devices. Furthermore, to take full advantage of such a device, a patient marker needs to be firmly mounted to a bone as a reference indicating the patient's location. And in order to find the position of the surgical tool relative to the patient and keep track of it, another marker also needs to be attached to the surgical tool, as shown in Figure 1.1. Fixing the patient marker does not expose the patient to life-threatening danger in most cases, but the damage caused to the bone is not in the patient's favour.

Hence, with the intention of overcoming the drawbacks of current surgical navigation systems and shortening the post-operation recovery time, considering the fact that interoperative fluoroscopies are wildly adopted, the idea of using fluoroscopic images as a medium for locating surgical tools has emerged. The pipeline can be roughly described as follows: first, a detachable reference frame (attachment piece) is to be designed for the surgical tool, and the position of the attachment piece needs to be found by using fluoroscopic images, the next step is to use the location of the attachment piece and its relative position with respect to the surgical instrument to locate

[1] https://ehealth.eletsonline.com/2012/10/apollo-performs-indias-first-navigation-surgery-for-maxillofacial-trauma/

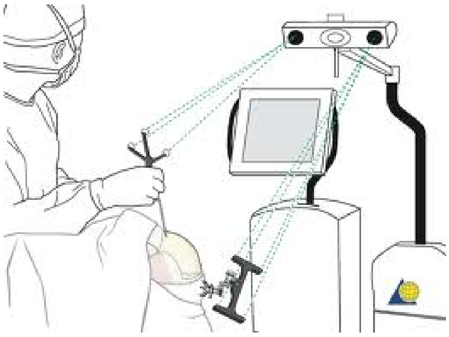


Figure 1.1: An simplified example of the use of existing surgical navigation systems[1]

the surgical instrument, then at the end of the pipeline, the surgical instrument's position with respect to the patient is calculated. This forms the basic idea of X23D, a currently ongoing project in the ROCS (Research in Orthopedic Computer Science) group from the Universitätsklinik Balgrist, which aims to provide surgical navigation through fluoroscopic devices.

This work focuses on exploring the potential solution for the first two parts of the pipeline: the development of a fluoroscopic-based surgical tool (attachment piece) pose estimation pipeline by detecting 2D coordinates of landmarks directly in radiographs. Through the development of the method, intraoperative fluoroscopy can be used to detect the attachment piece's 6D pose (translation + rotation), which is an essential component of any surgical navigation system.

To reach the final goal, firstly, an approach that can locate the attachment piece by utilizing the information that a single fluoroscopic image contains as much as possible needs to be found. For that purpose, we decide to explore convolutional neural networks, which are capable of extracting high-level features from images that are invisible to human eyes. There are three popular ways to find an object's 6D pose using 2D images, the first one is to perform regression directly on the rotation vector and translation vector. The second one is to make use of the depth information when it is available. And the third one requires a set of predefined 2D-3D corresponding landmarks. It first locates predefined landmarks in the 2D image using a neural network and then calculates the object's pose in the 3D world by feeding the coordinates of the corresponding points to the PnP algorithm. Due to the fact that fluoroscopic images are 2D images, it is not possible to obtain depth information from a fluoroscopy device. Thus the second approach cannot be used in this project. To make the most of the information that can be provided, namely the 2D images of the attachment piece, the 3D coordinates of predefined points (landmarks) on the attachment piece, and the corresponding 2D coordinates of the predefined points on 2D images, the third method is chosen to be applied in this project.

As one may be familiar, at the current stage, neural networks cannot guarantee an output with one hundred per cent accuracy. Moreover, the PnP algorithm cannot yield a definitive output as its accuracy increases along with the number of 2D-3D matching points. Therefore, it is also imperative to understand the importance of errors in 2D landmarks to the final 6D pose output by the PnP algorithm. Hence, the accuracy and stability of the PnP algorithm exclusively under this project setting and for this use case are explored. Based on the stability and reliability recognized, the performance of the model is assessed.

The structure of the thesis

- In Chapter 2, the current surgical navigation systems, as well as the commonly used image modalities, are elaborated. Then, the intention of providing surgical navigation and the decision to use fluoroscopy alone is justified by listing out the advantages and disadvantages of the current navigation systems.

- In Chapter 3, we first give an introduction to existing works on object 6D pose estimation. Next, the research that has been done to tackle the landmark detection task on 2D images is also discussed. And Popular neural network backbone structures are also briefly described in the last part of this chapter.
- In Chapter 4, the proposed approaches are detailed. From the design of the reference frame (attachment piece) to the generation of X23D-specific model performance expectation and image data, as well as the design of the network and post-processing pipeline.
- In Chapter 5, the implementation details and the experimental results are presented.
- In Chapter 6, the experimental results are further analysed, and the possible future working directions are discussed.
- In Chapter 7, the conclusion is drawn.

Chapter 2

Background Knowledge

2.1 Surgical Navigation

Surgical navigation technologies refer to computer aids that can provide surgeons with positioning information during operations. Resembling the GPS system, they deliver a map to surgeons indicating where they are and where the patient is. Not only do they allow surgeons to track the instruments' position and orientation with respect to the patient, but they also align the preoperative planning with the intraoperative images to offer guidance to the surgeons to help them pilot themselves through complex scenarios.

In this section, a brief introduction to the current navigation systems is given, followed by a short comparison and analysis of advantages and drawbacks, with the aim of clarifying the following three questions:

- 1. why are we doing surgical navigation?
- 2. why did we make the decision to use 2D-fluoroscopy?
- 3. why did we decide to achieve surgical navigation by using 2D fluoroscopy alone?

2.1.1 Intraoperative 2D-fluoroscopy Based

X-ray, the oldest form of medical imaging (Hatabu and Madore, 2021), has been widely used throughout the world since its first invention in 1895. To add flexibility to allow changing the angle from which the images were taken, Philips introduced the first mobile C-arm - a portable X-ray imaging device with the shape of a half moon.[1] The ability to rotate and move around the patient to take images from even extreme angles assists surgeons in various surgical procedures.[2] The rapid advancement of C-arm technologies, especially after 1957 when Philips attached an image intensifier to a monitor[1], made intraoperative image-based navigation achievable. A typical C-arm fluoroscopy system that is able to provide real-time X-ray imaging during interventions can be seen in Figure 2.1

In the year 2000, Foley et al. (2000) described a technology called "virtual fluoroscopy", where the C-arm fluoroscopy was combined with computer-aided surgery to magnify its advantages. Apart from the C-arm, this technology also involves two optical cameras fixed to one position during the entire surgery and light-emitting or retroreflective markers (see Figure 2.2), which can be tracked by the optical cameras (see Figure 2.3 left side). Generally, there are four steps that

[1] source:https://www.philips.com/consumerfiles/newscenter/main/shared/assets/Downloadablefile/FACT_SHEET_X-ray_history.pdf

[2] source: https://www.amberusa.com/blog/c-arm-fluoroscopy-and-image-aquisition

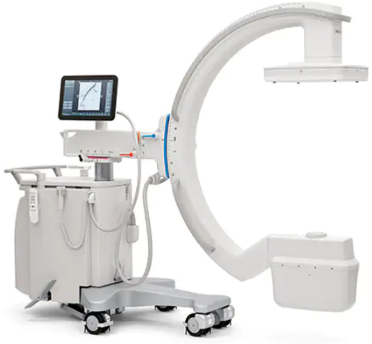


Figure 2.1: C-arm Fluoroscopy Example[3].

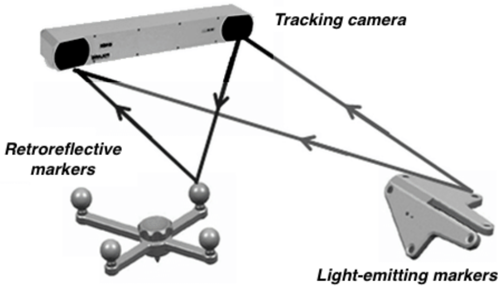


Figure 2.2: Retroreflective and Light-emitting Markers Świątek-Najwer et al. (2008)

compose the working flow for using this type of navigation system. The first step is to acquire one or more intraoperative images. Step two is to calculate the relative position of the C-arm and the patient by using a reference frame that is usually fixed to the patient's bone. The next step is to calibrate the acquired images. And in the final step, the virtual instrument is superimposed onto the images (see images on the right side of Figure 2.3), and the navigation starts from here (Foley et al., 2000).

2.1.2 Intraoperative 3D-imaging Based

Utilizing 3D images intraoperatively is another imaging option to help surgical navigation. 3D C-arm and intraoperative CT imaging are both able to provide surgeons with 3D images with different image qualities. Based on the 3D imaging systems, navigation systems were also developed. As in the 2D fluoroscopy-based navigation system, cameras and active or passive markers are needed to track the tool. The steps that are needed to use 3D-imaging-based navigation systems are similar to the necessary steps needed for using 2D-imaging-based ones described in the above section.

2.1.3 2D vs 3D Imaging

Instinctively, one may argue that 3D imaging-based navigation systems are superior and more helpful to surgeons compared to 2D fluoroscopy-based navigation systems due to their ability

[3] source: https://www.philips.com.au/healthcare/e/image-guided-therapy/mobile-c-arm/other

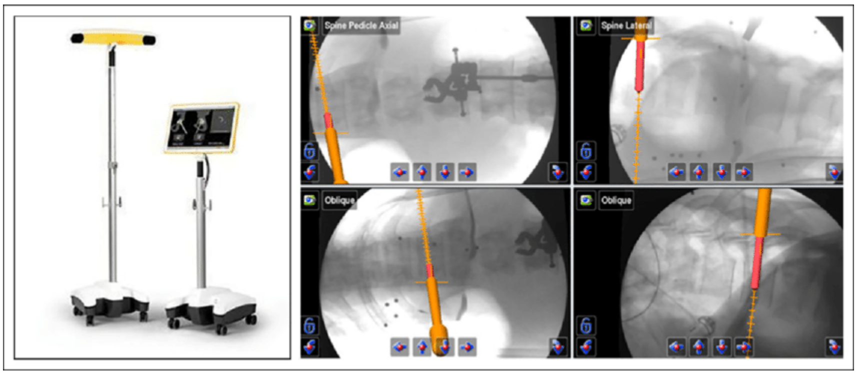


Figure 2.3: 2D FLUOROSCOPY BASED NAVIGATION SYSTEM HUSSAIN ET AL. (2020).

to render anatomical structures in 3D. Yet, Kułakowski et al. (2022) showed that using a 3D-fluoroscopy-based navigation system is not necessarily advantageous compared to using 2D-fluoroscopy in terms of accuracy. A similar conclusion has been drawn by Halm et al. (2020), they claimed that the use of 3D-fluoroscopy even prolongs the operating time with no noticeable improvement in the surgical results. Boudissa et al. (2022) also observed longer operating time as a result of using a 3D-imaging-based navigation system.

Another concern is the inevitable radiation exposure to both surgeons and patients. By Boudissa et al. (2022), greater radiation exposure to the patient has been associated with the use of a 3D-imaging-based navigation system. Mendelsohn et al. (2016) also agreed to this by stating they observed that 2.776 times radiation have been emitted to the patients compared to conventional fluoroscopy-guided cases. By contrast, according to Merloz et al. (2007), using 2D-fluoroscopy-based navigation systems could potentially eliminate the disadvantage of radiation exposure.

From this comparison, our second question has been answered. In brief, 2D fluoroscopy is a rational choice made after balancing operating time, surgical outcomes and radiation exposure.

2.1.4 Advantages and Drawbacks

Advantages

Comparing surgical navigation-guided surgeries and free-hand intraoperative imaging-guided surgeries, Madeja et al. (2022) have shown that the intraoperative radiation can be lowered by using a fluoroscopy-based 2D computer navigation system, and the surgical time can be shortened. Merloz et al. (2007), and Madeja et al. (2022) claimed that fluoroscopy-based 2D navigation systems are a safe, accurate and reliable method for placing screws during fracture osteosynthesis and in the lower thoracic and lumbar spine. Janssen et al. (2017), Hussain et al. (2018), Torres et al. (2012) and Navarro-Ramirez et al. (2017) showed that by using intraoperative 3D navigation systems, the safety, accuracy and reliability of surgery procedures and outcomes were all improved. Beyond that, the reliance on k-wires is also decreased (Goldberg et al., 2022).

By taking advantage of the now standard integrated image registration function provided with the navigation systems (Malham and Wells-Quinn, 2019), the registration of pre-operative images with intraoperative images give the surgeons a chance to find out the best surgical route outside the operating room without time pressure (Mezger et al., 2013), which can be cost-friendly to patients, and it may at the same time reduce the surgeons' brain workload during surgeries.

From the advantages of the surgical navigation systems stated above, our first question is answered. By using a surgical navigation system, surgeons have a chance to achieve optimal surgical results, minimize reoperation rate and reduce the likelihood of inducing postoperative complications (Otomo et al., 2022), thus saving money for the patients.

Drawbacks

There are certainly some drawbacks that come with the advantages.

First of all, the expense of incorporating intraoperative navigation systems. To be able to use such a system, an intraoperative imaging system needs to be established. There are several options, the most budget-friendly ones are 2D-fluoroscopy devices, a stand-alone 2D-fluoroscopy can cost from $25,000 to $70,000. Another option is to use 3D-fluoroscopy devices, which can cost from $300,000 to $1,000,000 per each. If the examination of soft tissue during the surgeries is needed, then CT scanners remain the only option. With the power of intraoperative CT scanners comes a hefty price tag, ranging from $600,000 to $1,200,000 for one. In addition to the expenses of the imaging system, the cost of the navigation system should not be underestimated. Optical tracking systems for spinal surgeries are priced from $250,000 to $700,000 (Malham and Wells-Quinn, 2019).

Another drawback of this type of system is the well-known "line-of-sight" issue. As mentioned before, cameras and markers are needed to keep track of the instruments. This means the markers need to be always visible to the cameras to perform the tracking task. Any block of direct sight from the cameras to the marks will lead to failure. Besides, infrared-based optical tracking systems which rely on the retroreflection from the markers are also prone to have multiple reflection issues due to the presence of other reflective surfaces in the operating room.

Another issue that inhabits intraoperative CT is that the imaging is not real-time. It is hard to be responsive to intraoperative alignment changes (Otomo et al., 2022). Moreover, a dedicated facility is required to set up such an imaging system, which takes up a noticeable amount of space in the operating room.

By listing some of the deep-rooted drawbacks of the optical tracking-based navigation system, the answer to our third question is also formed. X23D aims to bypass the "line-of-sight" problem and make the navigation system more accessible to low-budget medical facilities by lowering down not only the monetary requirement for establishing such systems but also the reliance on peripheral devices for achieving navigation.

Chapter 3

Related Work

With the development of neural networks in recent years, possible approaches to utilize the advantages of neural networks to achieve pose estimation of objects in the 3D world have emerged. In this chapter, the recent trends in using deep neural networks to perform object 6D (rotation + translation) pose estimation are discussed.

He et al. (2020) systematically evaluated the performance of non-learning-based approaches and learning-based (neural network-based) approaches on object pose estimation. According to the reported results, the accuracy of the non-learning-based approach was similar to that of the learning-based approach, yet learning-based methods outperformed traditional methods in terms of robustness. Despite the time-consuming training process and the heavy demand for computational power and storage space of the learning-based approaches, the short inference time of trained models is still advantageous compared to non-learning-based approaches. Therefore, we focus on using a neural network to accomplish our task. In the following sections, research has been done on learning-based pose estimation is introduced.

3.1 Learning-based Pose Estimation in 3D

This section discusses three current workflow fashions for using a neural network to assist the pose estimation.

Regardless of the heterogeneous models proposed in recent years, the overall design schemes can be summarized into: 1. The End-to-End fashion, where the models are trained to directly output the rotation and the translation with their 2D coordinates as input; 2. RGBD image-based fashion, where not only the RGB information is used, but also the depth information is integrated; 3. Two stages approach, where some predefined landmarks are first located on 2D images and then use the PnP algorithm to calculate the rotation vector and translation vector (He et al., 2020; Gamra and Akhloufi, 2021).

3.1.1 End-to-End Approach

Models designed following the end-to-end fashion (Figure 3.1) take 2D images as input and directly output the translation vector and rotation vector.

Hu et al. (2020) proposed a generic single-stage 6D object pose estimation network which can be combined with keypoint extraction networks and take RGB images as input to perform the object detection and 6D pose regression. The PoseNet (Kendall et al., 2015) is another well-known architecture to perform end-to-end pose estimation. It can alleviate the impact of difficult lighting and motion blur on the outcome and return the 6D pose by taking a single RGB image as input.

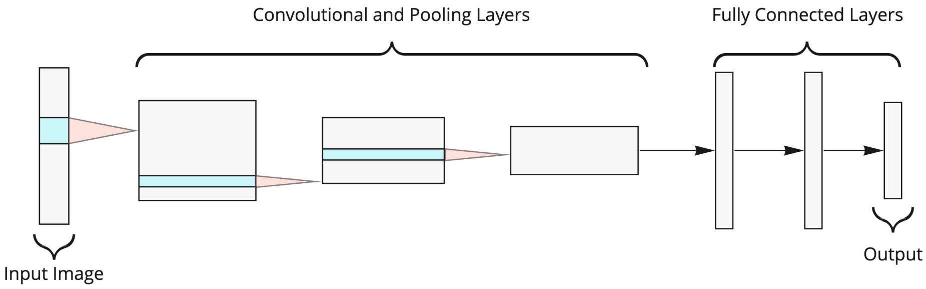


Figure 3.1: End-to-End approach design idea (Pachón et al., 2021)

Based on the PoseNet, Bui et al. (2017) developed X-ray PoseNet to learn a mapping between an object's simulated X-ray projection and its pose. A projector block has been added to reconstruct the 2D image based on the estimated pose output by the CNN model. The differences between the ground truth 2D image and the re-projected 2D image were added to the loss function and back-propagated to the CNN model to adjust the weights. RDpose (Presenti et al., 2022) also tried to use the X-ray image as input to output the object's pose. They separated the rotation task and the translation task right after ResNet-50 V2 (He et al., 2016b), but the task for the rotation branch was only to find the rotation around the vertical axis. They introduced two networks with similar architectures. One took a single X-ray image as input at a time, and the other took two X-ray images of the same object taken from different angles, and they observed a performance gain by adding the additional image. Xiang et al. (2017) proposed the PoseCNN to simultaneously estimate the pose of multiple objects in the scene. The translation vector was predicted by first putting a bounding box around the object and performing displacement vector regression to detect its centre, and then an estimation of distance was given. In the meantime, segmentation was also performed to help the centre localization. The final translation vector was calculated based on the camera's intrinsic parameters and the distance the displacement vectors provided. The rotation vector in quaternion form was also regressed based on the features within the bounding box extracted by the previous convolutional layers.

The Deep-6DPose (Do et al., 2018) performed detection, segmentation and 6D pose estimation tasks of multiple objects at the same time from a single RGB image. The estimation of the pose was based on the features extracted by using Mask R-CNN (He et al., 2017), and the segmentation head from Mask R-CNN was also adapted to predict a segmentation mask.

3.1.2 RGB-D Image-based Approach

The models mentioned in the End-to-End approach were all taking 2D images as input. The effect of using depth cues on the estimation of the object's pose was not fully explored. In this section, studies have been done on integrating depth information in predicting the pose are briefly introduced. Examples of depth information can be seen on the right side of Figure 3.2.

Guo et al. (2017) attempted to combine pre-trained PoseNet (Kendall et al., 2015) with LSTM blocks forming a pipeline to estimate camera pose from RGB-D image sequences. An advantage of their structure is its elasticity. The number of pipelines forming the network is flexible based on the complexity of the use case. They claimed that by doing such, to some extent, overfitting might be avoided. The DenseFusion introduced by Wang et al. (2019). tried to fully leverage the depth information by separating the input for the depth image from the RGB image. By embedding and

[1] source: https://www.v7labs.com/open-datasets/rgb-d-dataset

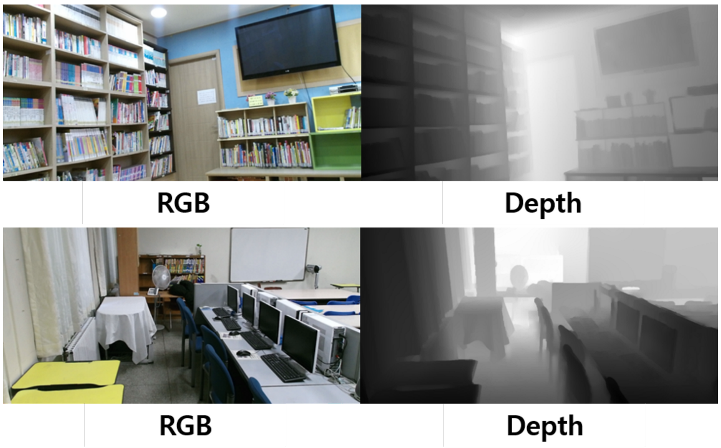


Figure 3.2: RGB-D Image Example [1]

fusing RGB values and point clouds at a per-pixel level, the model was able to benefit from the explicitly modelled local appearance and geometry information to better handle heavy occlusion scenes (Wang et al., 2019). The new MV6D (Duffhauss et al., 2022) tried to predict all objects in a scene by taking RGB-D images from multiple views. It separated the task-solving process into 3 phases. In the first phase, a PSPNet (Zhao et al., 2017) was used to extract features of each input RGB image from each view independently, and the depth information was extracted and combined, and the PointNet++ (Qi et al., 2017) was adopted to perform depth information feature extraction. At the end of the first phase, all the features were combined to feed into DenseFusion (Wang et al., 2019), and the output was then combined again with the previously extracted features to form point-wise feature vectors. In the second phase, based on the vectors from the first phase, 3D keypoint detection, 3D centre point detection and instance semantic segmentation were performed simultaneously. In the last phase, the least squares fitting algorithm was used to give out the rotation and translation.

3.1.3 Two-Stage Approach

Stage 1: 2D Feature Points Extraction

In the first step, models need to be trained to take 2D images as input, either output 2D coordinates of keypoints directly or output some features, from which the 2D coordinates of the keypoints can be extracted. The features can be heatmaps of the 2D coordinates, which encode the confidence of a pixel being one of the points of interest. Another option is to output displacement vectors, which represent the displacements from a pixel to a keypoint.

Stage 2: Calculate 6D Pose using PnP

The PnP stands for Perspective-n-Point. The PnP algorithm was first introduced by Fischler and Bolles (1981) (the detail will be introduced in Section 3.4) After that, it was used to estimate the pose of a calibrated camera with respect to the world/object coordinate system. It takes the points' 3D coordinates on objects in the world coordinate system and the corresponding points' coordinates on the 2D photo to make an estimation of the camera's 6D pose.

The BB8 (Rad and Lepetit, 2017) tried to predict the 2D coordinates of the 8 corners of 3D bounding boxes. They restricted the rotation range to mitigate the influence of the symmetry of the objects. They introduced a classifier to identify the range of the rotation of an object before feeding the image to the pose estimation network to decide whether an additional mirroring step is needed. Yet, evidence shows that BB8 suffers from a performance decrease when dealing with occlusions. Oberweger et al. (2018) used the sliding window technique to find the centre of the object. They claimed that by accumulating predicted heatmaps from several patches, the robustness of the model on handling occlusions could be improved. They extracted the global maximum from the average of the accumulated heatmaps to result in the final 2D locations, and then they used the PnP algorithm to calculate the final pose. Hu et al. (2019) also proposed to use of a patch-based method, where every grid of the visible part of an object was used to calculate displacement vectors from the centre of the grid to the keypoints to tackle the performance issue caused by occlusions. Besides this regression branch, the model also benefited from its segmentation branch. Li et al. (2019) introduced CDPN. They followed a partially end-to-end, partially two-stage fashion when designing the network, and they argued that the non-negligible differences between translation and rotation need to be taken into consideration. They separated the translation and the rotation estimation tasks after the feature extraction step. The translation branch performed the direct regression on the translation vector, while the final rotation vector was given by the PnP algorithm.

3.2 Keypoints Localization in 2D

In order to complete the two-stages approach, the first question that needs to be answered is, where are the key points on the image plane? To address this problem, methods have been developed in the field of 2D human pose estimation to detect the location of articulates. Since finding the keypoints of an object and finding the joints of humans in 2D images are interchangeable, the architectures developed to solve the 2D human pose estimation problem can be applied to locate keypoints of an object in 2D images.

Currently, as previously briefly mentioned, the models for 2D keypoints localization tasks can be classified into three groups: 1. direct regression on 2D coordinates, 2. regression on displacement vector, 3. regression on the Gaussian heatmaps generated around the keypoints' 2D coordinates. These three methods are described in detail in the following sections.

3.2.1 Direct Coordinates Regression

As the name of this group suggests, the form of the output of networks that perform direct regression on coordinates is the exact location of the keypoints in a 2D image. A shared idea behind designing this type of network is that after an image-specific feature extraction step performed by the convolutional layers, several fully connected layers then give in the 2D coordinates of the keypoints by consuming the extracted information.

In human joint detection, DeepPose (Toshev and Szegedy, 2014) was introduced. The model married the recurrent idea with a convolutional neural network. Instead of having only one stage, the authors reused the feature extraction plus fully connected block by stacking them together to achieve a zoom-in effect. After having the coordinates values in coarse-scale yielded by the first stage, the subsequence stages predicted the displacement from the previous stage output to the ground truth. After the last stage of the network, linear regression was performed to predict the 2D coordinates. The authors argued that by having a cascade structure in the network, the details of the images were able to be revealed, which exposed the network to higher-resolution images and allowed the network to perform stage-by-stage refinement to result in higher precision.

In the medical field, Riegler et al. (2015) proposed a relatively simple network to predict landmarks' location in synthetic MR/CT images. The feature extraction part of the network was composed of only three convolutional layers and two maxpooling layers. They showed that the compromise of using synthetic data could be beneficial when faced with insufficient training data problems in the medical domain. The L model in Laina et al. (2017) adopted the same idea and used ResNet50 (He et al., 2016a) as the backbone but focused on detecting landmarks on surgical instruments in real-time videos. The SL model from the same research (Laina et al., 2017) attached a segmentation branch to the L model to predict a mask of the target object and the 2D coordinates of the landmarks at the same time. By sharing the encoder, the loss from the segmentation branch was able to have an implicit influence on the regression branch, and a slightly improved result was observed.

3.2.2 Displacement Vector Regression

In this group, the model tries to predict the vector pointing to all landmarks from any pixel or patch. As the name of the vector suggests, it represents the displacement between a pixel or patch and a landmark. Not only does the displacement vector contain distance information, but it also contains direction information.

In order to alleviate the impact of the shortage of medical image data, a two-stage task-oriented network (T^{2}DL), which was able to perform simultaneous landmark detection in real-time, has been proposed by Zhang et al. (2017). This network had two stages focusing on two separate tasks. In the first stage, instead of using a complete image, the network took image patches as input. The relationship between patches and the displacement vectors from patches to landmarks was learned. In the second stage, more layers were added to the first network in order to emphasize the importance of neighbouring patches to the ground truth locations, and in this stage, displacements were used as a base for the final 2D coordinates output. The learned network weights were also used in the second stage to make full use of the patch associations learned in the first phase. Noothout et al. (2020) made an attempt to use displacement vector regression to predict landmarks' location not only in coronary CT angiography scans but also in olfactory MR scans and cephalometric X-rays in 2 stages. The basic structure of their network used slightly modified ResNet (He et al., 2016a) as the backbone, and two heads were added to tackle the displacement vector regression task and determine whether a patch contained landmarks. In the first stage, the basic structure was used to perform global image analysis. After the first stage, the patches predicted to contain landmarks were used as input for the second stage, and a smaller version of the basic structure was reused in the second stage to process the zoomed-in image to yield more accurate final landmark locations.

3.2.3 Heatmap Regression

In this section, models performing landmark localization through heatmap regression will be briefly introduced. Heatmaps used for model training are multivariate Gaussian distributions centred around the ground truth 2D landmark coordinates. The closer the pixel is to the ground truth coordinates, the higher the probability that it is ground truth. Hence, the heatmaps also implicitly encode the distance from a pixel to the ground truth coordinates. As shown in Figure 3.3, on the left side of the figure, a Gaussian heatmap in 2D is visualized, and On the right side is the corresponding 3D image.

Payer et al. (2016) proposed SpaticalConfiguration-Net architecture to find landmarks in 2D and 3D medical images. The architecture consisted of 3 blocks. The first block containing three convolutional layers was used to generate local appearance heatmaps for the landmarks. Afterwards, the spatial configuration blocks they introduced were used to learn the relative position

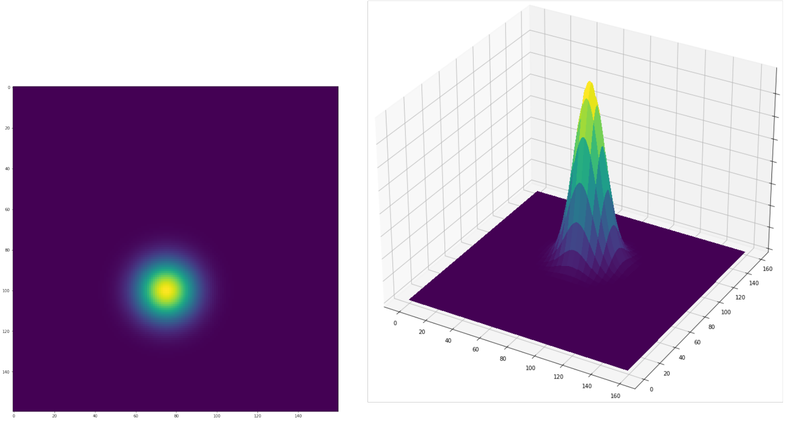


Figure 3.3: 2D Gaussian Heatmap and 3D Gaussian Heatmap

between a landmark and remaining landmarks. By adding all intermediate heatmaps that represent spatial configuration between landmarks, a low-resolution heatmap that indicates the possible location of a landmark was generated. In the final block, the low-resolution heatmap and the local appearance heatmaps were added to resolve the ambiguity that local appearance heatmaps introduced. The idea of achieving the magnification effect through global and local stages was adopted by Zhong et al. (2019). In the global stage, an encoder-decoder structure was used to generate coarse attention. In the local stage, patches were extracted based on the attention from the prior stage. The same encoder-decoder structure was used to predict fine-grained heatmaps from patches. The final prediction was given by merging the heatmap patches.

Wei et al. (2016) introduced the Convolutional Pose Machine. Despite the fact that it was initially used to detect human articulates, this successful network structure can also be transferred to detect landmarks in the 2D medical images. Similar to DeepPose (Toshev and Szegedy, 2014), the Convolutional Pose Machine was built with multiple convolutional layers in recurrent style. Unlike DeepPose, which outputted 2D coordinates directly, Convolutional Pose Machine yielded a belief map (heatmap) for each landmark. In each stage, a shared intermediate supervision block which took the original image as input and was composed of convolutional layers and maxpooling layers, was introduced to deal with the vanishing gradient problem. With the deepening of the network, the receptive field of the network was also enlarging. The author also proved that this enlarged receptive field was beneficial to the accuracy of the network. Another advantage was that the network was able to deal with images from arbitrary views. Bier et al. (2018) applied this network to solve the view-independent anatomical landmark detection task in X-ray images. Although the network was trained on synthetic X-ray images, the trained model was still able to directly process real X-ray images and output promising results.

Another frequently used structure is the U-Net, which was initially proposed by Ronneberger et al. (2015) to perform segmentation tasks on medical images. The detail of its structure will be presented in the next section. Works inspired by the U-Net on tackling the landmark localization task have also achieved delightful results.

The Concurrent Segmentation and Localization (CSL) model introduced by Laina et al. (2017) has gained satisfying success when performing real-time surgical instrument tracking in minimal invasive surgery. The network structure was a merge of U-Net (Ronneberger et al., 2015) and a Fully Convolutional Network. As in Noothout et al. (2018); Hu et al. (2019), the model also had a segmentation branch to help the localization branch. The networks were constructed with an

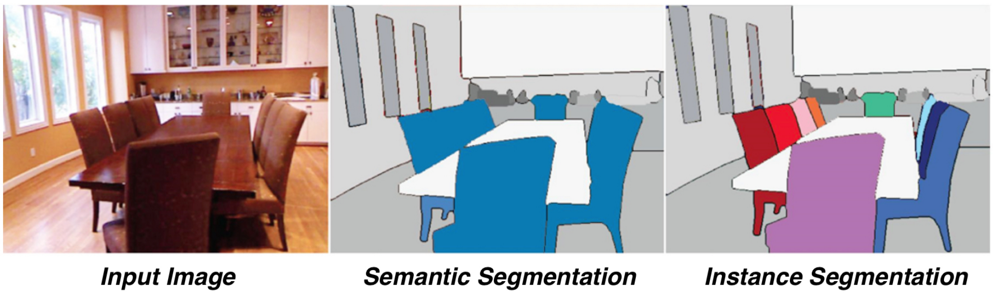


Figure 3.4: Examples of Segmentations (Silberman et al., 2014)

encoder block and a decoder block. The encoder block was composed of ResNet50 (He et al., 2016a) and an extra residual block. As in the U-Net (Ronneberger et al., 2015), the decoder part was composed of several upsampling blocks and skip connections from the encoder, passing low-level but high-resolution information from the encoder to the decoder. The network was split into two branches near the end of the structure. One of which was assigned with segmentation task, and the other one kept focusing on localization. Afterwards, the segmentation output and the convolution output were concatenated to produce the final heatmaps. The concatenation enabled the segmentation branch to provide a guide to the localization branch.

Research has also been done on exploring whether using a single U-shape network to predict heatmaps for landmarks was already sufficient to reach pleasant outcomes. Kang et al. (2021) examined the ability of U-Net (Ronneberger et al., 2015), and U-Net with attention gate (Oktay et al., 2018) on handling perturbed X-ray images. Their results confirmed that a single U-shaped network could adequately handle the heatmap regression task to some extent. They also observed a performance gain by perturbing images. Fard et al. (2022) redesigned the kernel size used in the U-Net. The altered structure proved to be useful when detecting landmarks in spine X-ray images.

3.3 Segmentation in 2D Images

A trend of using segmentation to help with landmark localization can be seen in the previously introduced models. The focus of this section is to give a brief introduction to three popular network architectures that are wildly adopted as backbones when performing medical image-related segmentation tasks.

With segmentation, models are trying to distinguish objects from the background or even separate different objects from each other. Figure 3.4 serves as an example and provides a visual interpretation of segmentation tasks. Semantic segmentation tries to categorize items into different categories, whereas instance segmentation aims to distinguish instances from instances.

Fully Convolutional Network The fully Convolutional Network introduced by Long et al. (2015) achieved state-of-art semantic segmentation results back in 2015. The key idea was to have a network composed of only convolutional related layers that can take images with arbitrary sizes and produce sized images that indicate the segmentation of the object in the images. An advantage of having no fully connected layer is that it enables the model to learn a pixel-wise representation of the original input image. As the network goes deeper, the deep features are obtained, but the spatial information gets lost easily. To overcome this limitation, the authors introduced skip connections to pass low-level features containing more location information to later layers, and experiments also confirmed the effectiveness of this information-preserving design decision. The authors also showed the method to convert existing CNN models with fully connected layers to a fully convolutional network. By applying "convolutionalization" which is “convolutions with kernels that cover the entire input regions” (Long et al., 2015), the dense layers can be easily transformed to a fully convolutional style. They also observed a faster reference speed with either the original Fully Convolutional Network or convolutionalized networks.

U-Net Inspired by the Fully Convolutional Network (Long et al., 2015), Ronneberger et al. (2015) built the U-Net, one of the most famous segmentation networks in the biomedical field. It integrated encoder-decoder structure, fully convolution idea and skip connections into one network. Compared to FCN, U-Net had more feature channels in the expansion path, which empowered the flow of the context information to high-resolution layers. The skip connections in each layer ensured the high-resolution spatial information from the contracting path could be retained.

ResNet He et al. (2016a) proposed the ResNet in 2015 for image recognition. It reached state-of-art performance in classification, object detection and object localization. Itself and its variants are still wildly employed until now. The tendency that the deeper the network, the better the performance has made researchers willing to work on the depth of their networks. But findings (He et al., 2016a; He and Sun, 2015) suggest that there is a chance of counterintuitive accuracy degradation when the network goes deeper. ResNet’s first and foremost element, the residual connection, was designed to fight the performance decrease caused by deeper networks. Residual blocks, consisting of layers of convolution operations and a residual connection from block input to block output, were stacked on top of each other to form the final ResNet. Evidence from He et al. (2016a) shows that with shortcut connections, even in the form of identity mapping, the notorious performance degradation problem could be solved.

The above architectures can often be seen in research (Laina et al., 2017; Presenti et al., 2022; Gao et al., 2019; Kang et al., 2021; Fard et al., 2022; Ding et al., 2021; Lu et al., 2021), either individually or in combination, as cornerstones for solving medical image-related tasks.

3.4 PnP Algorithm

As briefly mentioned in Section 3.1.3, the PnP algorithm is able to provide us with the camera’s pose in the form of a rotation vector and translation vector. The resulting rotation matrix and translation vector, which form the camera’s extrinsic parameters, can be used to bring an object’s position from the world coordinate frame to the camera coordinate frame. Together with the camera’s intrinsic parameters, namely, focal length and focal point, the object can be projected onto the image plane. Figure 3.5 shows a simplified workflow of this algorithm. The inverse of the camera’s movement is equivalent to the movement of the object in the same coordinate system around the same rotation centre. Thus, this algorithm is commonly used to estimate the object’s/camera’s pose.

Before, the "PnP algorithm" notion was slightly abused, as it was used to refer to the family of algorithms that solves the Perspective-n-Point problem and finds the camera’s pose. From that family, the Efficient PnP algorithm (Lepetit et al., 2009) (which will be referred to as the PnP algorithm again in later sections) is selected to provide assistance in locating the camera/object. It is acknowledged that when using the PnP algorithm, it is always preferred to have as many point-pair as possible to increase accuracy (Lu, 2018). However, with more points involved, the computation complexity increases at the same time. Hence, balancing accuracy and computation

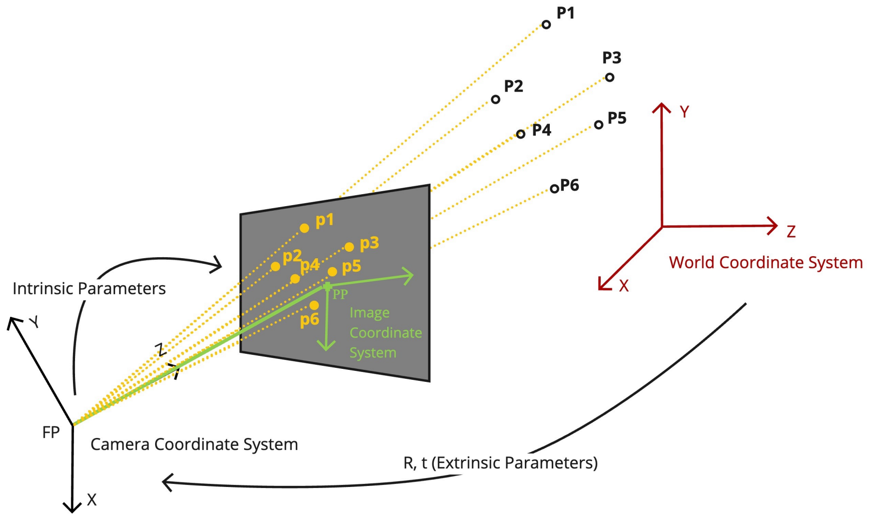


Figure 3.5: PnP Algorithm

complexity is needed as both factors are of great importance. The EPNP algorithm has a promising complexity of $O(n)$ where n is the number of matching keypoints pairs, and it creates four virtual points around each 3D point, which can improve the algorithm's stability, thereby increasing the algorithm's accuracy. For this reason, we preferred to use the EPNP algorithm over the rest of the family.

Chapter 4

Methodology

4.1 The Design of the Attachment Piece

During the surgeries, an assortment of surgical instruments are involved, and the type of instruments chosen highly depends on the patient-specific settings and varies from patient to patient. Thus, in order to be able to accommodate different kinds of surgical operations inspired by the instrument markers used in traditional optical surgical tool tracking systems, a structure that is easily distinguishable from surgical instruments is required to accommodate various surgical settings. The attachment piece is required to be separable from surgical instruments either in the real world or in imaging systems, attachable and detachable, in a reasonable size and able to maintain its shape regardless of the instrument it is being attached to.

Thanks to the engineers from the ROCS group, a 3D prototype of the attachment piece in a unique shape has been designed. In Figure 4.1, images of the 3D rendering of the attachment piece from its front view, lateral view, top-down view and a random view are displayed. As can be seen in Figure 4.1, the attachment piece has three branches, the main branch with a sphere, a left branch with another sphere in a smaller size and a right branch with a hemisphere. The idea of having spheres is closely related to the landmark selection. An advantage of spheres is that in 3D, the relative location of the sphere's centre to its surface is always stable. And the distance from the surface to the centre is always equal to the radius of the sphere, irrespective of the view. This can also be extended to its 2D projection, the centre of the circle is fixed with respect to the circumference. Having a hemisphere is intended to add variety to the shape while maintaining a degree of constant relative position.

From the top-down and lateral views, an attempt to avoid coplanar and collinear branches can be observed. The difference in the angles between the main branch and the sub-branches, the shape difference between the tops of the two sub-branches, the positional difference of the connection point of the sub-branch to the main branch, the thickness difference of the three branches, and the curvature difference of the sub-branches make the structure distinct from commonly used surgical instruments, and its asymmetry can indicate its rotation to some extent. The initial decision to 3D print the structure in metal or plastic ensured that it would be rigid. The size of the attachment piece is negotiable with 3D printing, which allows us to search for the optimal size balancing the interference it may cause to the surgical procedures and the visibility of it in the imaging systems.

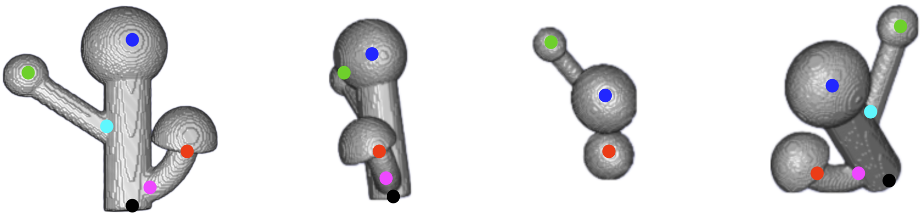


Figure 4.1: 3D Views of the Attachment Piece from Different Angles

4.2 Landmark Selection

Following the decision to use the two-stage approach to accomplish the pose estimation task, a crucial and prerequisite step is to place the landmarks, which are going to be used in calculating the final pose through the PnP algorithm, in appropriate locations on the 3D model.

It only takes 3 points for the PnP algorithm to calculate a possible transformation between the camera coordinate system and the world coordinate system, however, the result might not be exclusive, yet fortunately, it can provide us with a solution space with a limited amount of solutions (Acuna and Willert, 2018). Under our application settings, having multiple potential solutions is not an option. It is compulsory to restrict the number of solutions down to one. In order for the PnP algorithm to result in a unique solution, an extra point is demanded to filter the solution space, which means that at least 4 points are required. Meanwhile, it is a consensus that increasing the number of matching points is approximately equivalent to increasing the algorithm's resilience when facing the noise that might exist in 2D coordinates of the matching points on the image plane (Acuna and Willert, 2018). Having the above mathematical knowledge and empirical evidence in mind, it is rational for us to not only meet the minimum requirement of 4 matching points but also add more points to seek compensation when noise is present.

As a result, taking the characteristics of the shape of the prototype into consideration, 6 landmarks were manual-selected on the 3D prototype. As illustrated in Figure 4.1, two of the landmarks are located right in the centre of the spheres (marked in blue and green), and two of the landmarks are located at the places where the main branch and the sub-branches connect (marked in cyan and pink), one is on the bottom of the hemisphere (marked in red), and the last one is on the bottom of the main branch (marked in black). Furthermore, what needs to be remarked is that all the landmarks have different depth values (Z values), and any of the three points cannot be lined up using a straight line. The purpose of having this setting is to avoid issues that collinearity and coplanarity may introduce when calculating pose using the PnP algorithm.

An additional factor that may require attention is that owing to the reason that high network performance is desired, it is preferred if one can ease the network's job of learning and inferring landmarks' locations by manipulating the location of the landmarks at the designing phase. So, as can be seen in Figure 4.2, in the 2D projections of the model, the location of the blue and the green landmarks remain relatively stable, and their distances to the surrounding circles are irrelevant to the viewing angle. For the cyan, red, pink, and black landmarks (in the third image, the black landmark is marked in grey to be differentiable from the tool itself), the intention is to set them in places encircled by complex textures. We hypothesize that the above-mentioned features and the angles between the main branch and the sub-branches, the hemisphere of the right branch and its limb, and the colour shifting at the bottom of the tool could potentially be strong hints indicating where the landmarks are for the model.

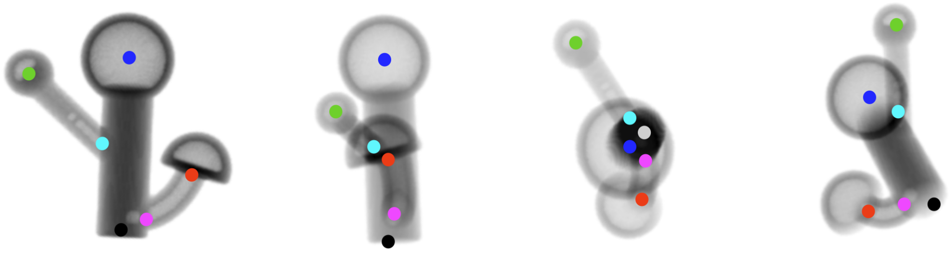


Figure 4.2: 2D Views of the Attachment Piece from Different Angles

4.3 Model Performance Expectation Generation

To assess the model's performance, the importance of the Euclidean distances between the prediction and the ground truth on 2D images to the 3D final pose of the attachment piece needs to be investigated. Hence, the error simulation process aiming to identify an upper bound for our initial model expectation (error tolerance) is designed. Before introducing our error simulation process in detail, we present the fundamental facts supporting our pipeline.

Combined with the description in Chapter 3, the rotation and translation applied to the camera when keeping the object unmoved can be inversely applied to the object while keeping the camera at its original position. The resulting images are the same as long as the rotation and translation are with respect to the same coordinate system and around the same rotation centre.

Figure 4.3 contains vivid demonstrations of the above-mentioned process (omitting translations). Assume that the attachment piece and the C-arm are both at their initial position in the first image, respectively. The rotation centre is the centre of the attachment piece, which is marked out with a black square on the image, and the coordinate system obeys the right-hand rule with Z-axis pointing towards us. By taking a picture under the initial pose configurations, we would obtain an X-ray image of a lateral view of the attachment piece similar to Figure 4.4(a) (the method used to generate the X-ray images will be introduced in Section 4.4.1). In the second image of Figure 4.3, if we rotate the attachment piece around the Z-axis 180 degrees counter-clockwise, meanwhile keeping the C-arm still, the tool would be in an upside-down pose, and the resulting X-ray image would be similar to Figure 4.4(b) which containing a lateral view of the upside-down attachment piece. By only rotating the C-arm 180 degrees clockwise and leaving the attachment piece with its initial pose, as shown in the third image of Figure 4.3, an X-ray scan with the same view as the X-ray scan taken under the pose configuration of the second image can be generated (see Figure 4.4(c)).

Based on this observation, we argue that the translation and rotation vectors given by the PnP algorithm after moving the attachment piece alone should be the same as the translation and rotation that is used to calculate the camera projection matrix after moving the camera alone (i.e., they both represent the transformation required to express a point in the camera coordinate system). These findings allow us to form our initial model performance expectation (acceptable error upper bound) generation steps into 3 phases. The first phase simulates the movement of the attachment piece alone and obtains the 2D coordinates of the landmarks. The second phase simulates the movement of the C-arm to generate the ground truth camera pose. The reason for having the second phase will be explained at the end of this section. In the last phase, errors are introduced to the 2D coordinates obtained in the first phase to mimic the behaviour of the

[1] Adapted from Ritschl et al. (2015)

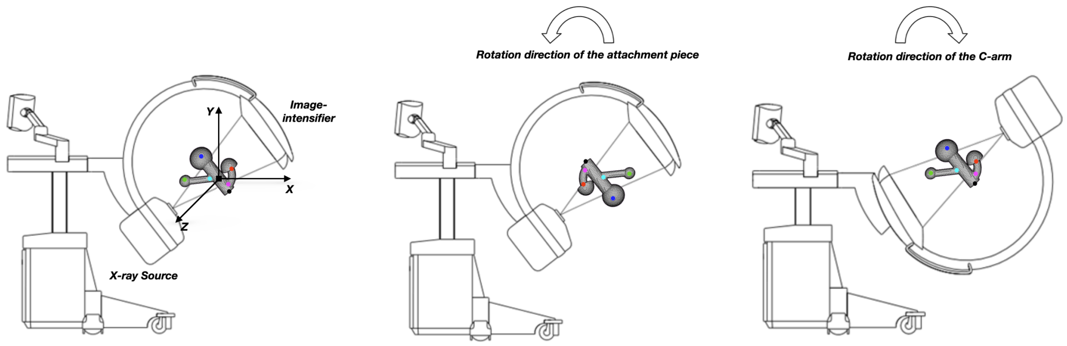
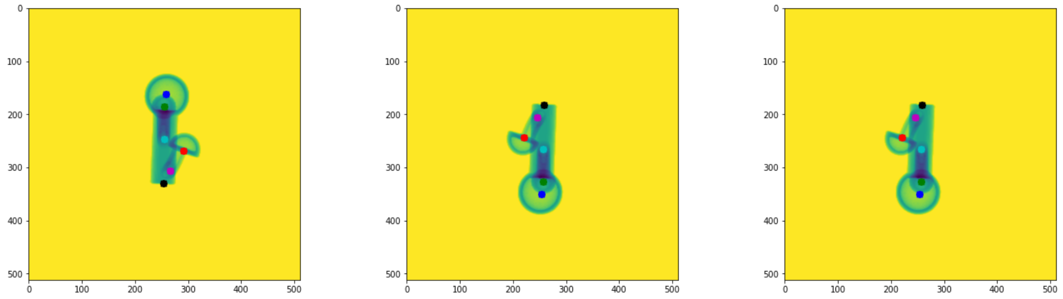

Rotation direction of the attachment piece
Rotation direction of the C-arm
Image-intensifier
Y
X
Z
X-ray Source

Figure 4.3: Movement of the C-arm and the Attachment Piece [1]



(a) Assumed initial pose X-ray scan

(b) Rotate the attachment piece 180 degrees counter-clockwise

(c) Rotate the camera 180 degrees clockwise

Figure 4.4: Example X-ray scans before and after rotating the attachment piece and the C-arm

neural network (i.e., the error that the predictions of the model may possess), and the 2D coordinates with error introduced are used to calculate camera pose using the PnP algorithm, then the comparison between the ground truth and the output of the PnP algorithm is performed.

What needs to be borne in mind when walking through the following steps is that the 3D coordinates of the landmarks in the world coordinate system L_{3d} are already delivered after the landmark selection step, which is also a prerequisite for the use of the PnP algorithm. And the camera intrinsic parameters (i.e., the initial location of the focal point, the focal length, and the location of the principal point on the image plane) are already known. The detailed procedures are as follows:

Phase 1: Attachment Piece Moving and 2D Coordinates Acquiring

1. Set an intended rotation $\theta = (\theta_x, \theta_y, \theta_z)$ and translation $t = (t_x, t_y, t_z)$ that are going to be applied to the camera. And the movement to the attachment piece is the inverse of the movement of the camera.

2. Calculate the rotation matrix R based on the angles set in step 1. The rotation matrixes around X, Y and Z-axis, R_x, R_y and R_z, as well as their expressions in homogeneous coordinates R_x^H, R_y^H and R_z^H can be calculated using Equation (4.1), (4.2) and (4.3) respectively.

After obtaining R_x, R_y, R_z and R_x^H, R_y^H, R_z^H, the final rotation matrix R and its expressions in homogeneous coordinates R^H are calculated using Equation (4.4).

$$R_x(\theta_x)_{3\times3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x \\ 0 & sin\theta_x & cos\theta_x \\ 0 & 0 & 0 \end{bmatrix} R_x^H(\theta_x)_{4\times4} = \begin{bmatrix} R_x(\theta_x)_{3\times3} & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \tag{4.1}$$

$$R_y(\theta_y)_{3\times3} = \begin{bmatrix} cos\theta_y & 0 & sin\theta_y \\ 0 & 1 & 0 \\ -sin\theta_y & 0 & cos\theta_y \\ 0 & 0 & 0 \end{bmatrix} R_y^H(\theta_y)_{4\times4} = \begin{bmatrix} R_y(\theta_y)_{3\times3} & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \tag{4.2}$$

$$R_z(\theta_z)_{3\times3} = \begin{bmatrix} cos\theta_z & sin\theta_z & 0 \\ -sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} R_z^H(\theta_z)_{4\times4} = \begin{bmatrix} R_z(\theta_z)_{3\times3} & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \tag{4.3}$$

$$\begin{aligned} R_{3\times3} &= R_x(\theta_x)_{3\times3} R_y(\theta_y)_{3\times3} R_z(\theta_z)_{3\times3} \\ R_{4\times4}^H &= R_x^H(\theta_x)_{4\times4} R_y^H(\theta_y)_{4\times4} R_z^H(\theta_z)_{4\times4} \end{aligned} \tag{4.4}$$

3. Generate the translation matrix based on the translation vector set at step 1. This step is a relatively straightforward step. The translation matrix T^H in homogeneous coordinates can be obtained by plugging in the translation vector to Equation (4.5).

$$T^H = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

4. Calculate the final transformation matrix A using Equation (4.6).

$$A = T^H R^H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

5. Calculate the 3D landmarks' coordinates in homogeneous coordinates after the transformation using Equation (4.7).

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = A^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.7}$$

6. Generate the camera projection matrix using Equation (4.8). The projection matrix generated at this stage is essentially the initial configuration of the camera, as no C-arm movement is yet involved.

   The definition of the K matrix and its expression in homogeneous coordinates K^H can be seen in Equation (4.9). They stand for the camera's intrinsic parameters and encode the focal length and the principal point. f_x and f_y in Equation (4.9) represent the focal length. And C_x and C_y are the location of the principle point on the image plane, which is the foot of the perpendicular from the camera lens to the image plane, as the green point PP shown in Figure 3.5. The X_0 in Equation (4.8) represents the 3D location of the focal point (which can be roughly seen as the location of the camera).

$$P_{3\times4} = K^H A = [KR| \; - KRX_0] \tag{4.8}$$

$$K = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} K^H = \begin{bmatrix} f_x & 0 & C_x & 0 \\ 0 & f_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.9}$$

7. After obtaining the projection matrix from step 6, we use it to project the attachment piece to the image plane using Equation (4.10). The corresponding 2D coordinates L_{2d}^H of the 3D landmarks on the image plane in homogeneous coordinates can also be calculated using the same equation. By removing the last row of L_{2d}^H, the ground truth 2D coordinates of the landmarks L_{2d} can be acquired.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P_{3\times4} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.10}$$

Phase 2: C-arm Moving

8. Apply the rotation matrix R and translation vector t to the camera using Equation (4.11). This step is to move the focal point X_0 of the camera according to the transformation matrix generated at step 4.

$$X_{0_{new}} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + R_{3\times3}X_0 \tag{4.11}$$

9. Calculate camera's projection matrix P by replacing the X_0 with $X_{0_{new}}$ in Equation (4.8).

10. Define the upper left 3×3 submatrix of the projection matrix $P_{3\times4}$ as $M_{3\times3} = P_{ij}$ where $1 \leq i \leq 3$ and $1 \leq j \leq 3$. Use QR decomposition to decompose matrix $M_{3\times3}$ into an orthogonal matrix and an upper triangular matrix, and these matrices are the rotation matrix R_{qr} and the camera intrinsics K we are looking for.

11. Having matrix $M_{3\times3}$ and $P_{3\times4}$ at hand, based on the implicit message from step 10 and Equation (4.8) where $M_{3\times3} = KR_{qr}$, and $P_{ij} = -KR_{qr}C, 1 \leq i \leq 3, j = 4$ where C stands for camera centre, and it can be recovered using Equation (4.12).

$$C = -M_{3\times3}^{-1} \begin{bmatrix} P_{14} \\ P_{24} \\ P_{34} \end{bmatrix} \tag{4.12}$$

12. The translation vector t_{qr} can then be restored using Equation (4.13).

$$t_{qr} = -R_{qr}C \tag{4.13}$$

Phase 3: Error Analysis and Performance Expectation Generation

13. Generate errors for the 2D landmarks with the following steps.

    (a) Generate 6 unit vectors $\hat{e_i}, 1 \leq i \leq 6$, which will be applied to the ground truth 2D coordinates, with random directions using Equations in (4.14).

    $$\begin{aligned} v_i &= \begin{bmatrix} v_{i1} \\ v_{i2} \end{bmatrix} \sim \mathcal{N}_2(\mathbf{0},\, \mathbf{1}) \\ ||v_i|| &= \sqrt{{v_{i1}}^2 + {v_{i2}}^2} \\ \hat{e_i} &= \frac{v_i}{||v_i||} \end{aligned} \tag{4.14}$$

    (b) Use the unit vectors to form an error matrix:

    $$Err = \left[\hat{e_1}, \hat{e_2}, \hat{e_3}, \hat{e_4}, \hat{e_5}, \hat{e_6}\right]^T \tag{4.15}$$

    (c) Final error E_f is calculated by the multiplication operation (4.16) of the error matrix Err and a given scalar s_e. s_e is used to alter the magnitude of the unit vectors.

    $$E_f = s_e \cdot Err \tag{4.16}$$

14. Introduce the error generated to the ground truth 2D landmarks' coordinates L_{2d} by an addition operation (4.17).

    $$L_{2d}^{err} = (L_{2d}^T + E_f)^T. \tag{4.17}$$

15. Plug in the 3D coordinates L_{3d} of the landmarks and the compromised 2D coordinates L_{2d}^{err} into the PnP algorithm to calculate the rotation vector r_{pnp} and the translation vector t_{pnp}, and convert the rotation vector r_{pnp} into rotation matrix R_{pnp} using Equation (4.1), (4.2), (4.3) and (4.4).

    What to be noted here is that the rotation vector and the translation vector acquired from the PnP algorithm are the rotation and the translation with respect to the world coordinate system, they are not necessarily equivalent to the rotation and translation set at step 1. In the case where the movements of the attachment piece and the C-arm are around a rotation centre other than the origin of the world coordinate system, the r_{pnp} and the t_{pnp} are most likely different from the transformation set at step 1.

16. Convert the ground truth rotation matrix R_{qr} and R_{pnp} into quaternion form using equations in (4.18) resulting in corresponding quaternions q_{qr} and q_{pnp}, which are substitutes for the Euler angles and rotation matrices.

    $$\begin{aligned} q_r &= \frac{1}{2}\sqrt{1 + R_{qr_{11}} + R_{qr_{22}} + R_{qr_{33}}} \\ q_i &= \frac{1}{4_q r}(R_{qr_{32}} - R_{qr_{23}}) \\ q_j &= \frac{1}{4_q r}(R_{qr_{13}} - R_{qr_{31}}) \\ q_k &= \frac{1}{4_q r}(R_{qr_{21}} - R_{qr_{12}}) \\ q &= (q_r, q_i, q_j, q_k) \end{aligned} \tag{4.18}$$

17. Calculate the rotation error using Equation (4.19).

$$\begin{gathered} E_R = 2 \cdot arccos(|< q_{qr}, q_{pnp} >|) \\ < q_{qr}, q_{pnp} >= q_{qr_{11}} q_{pnp_{11}} + q_{qr_{22}} q_{pnp_{22}} + q_{qr_{33}} q_{pnp_{33}} + q_{qr_{44}} q_{pnp_{44}} \end{gathered} \tag{4.19}$$

18. Calculate the ADD error (Hinterstoisser et al., 2012) using Equation (4.20), which is the average distance between the reconstructed 3D landmarks using R_{pnp}, t_{pnp} and the ground truth 3D landmarks calculated using R_{qr} and t_{qr}. X is the set that contains all the 3D landmarks, $n = |X|$ is the number of landmarks on the attachment piece, and x is the 3D coordinate of a single landmark.

$$ADD = \frac{1}{n} \sum_{x \in X} ||(R_{pnp} x + t_{pnp}) - (R_{qr} x + t_{qr})||_2, \tag{4.20}$$

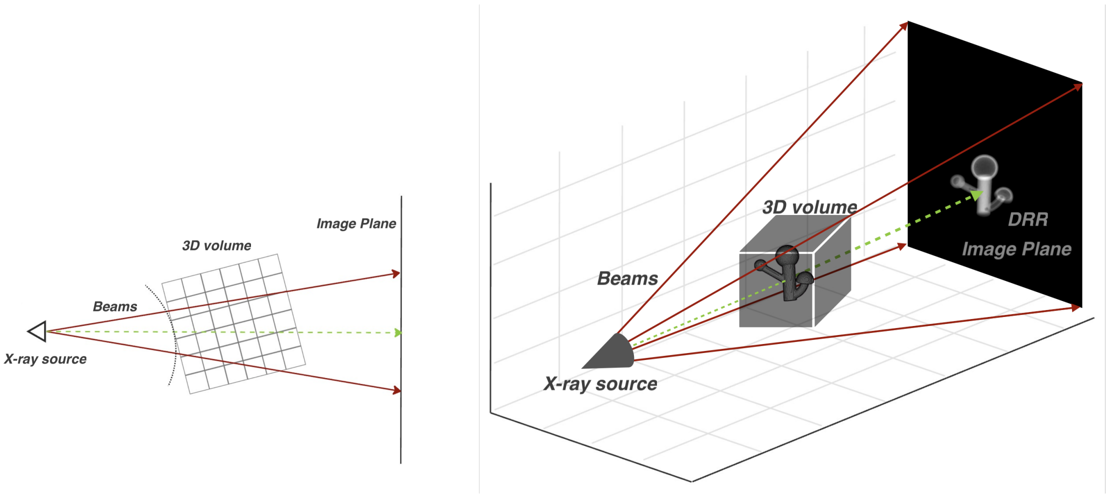
19. Repeat step 13 to 18 with different error vector magnitude s_e to investigate the impact of the 2D error on the rotation matrix R_{pnp} and the reconstruction under the specified rotation setting $(\theta_1, \theta_2, \theta_3)$, then report the rotation error E_R and the ADD error.

20. Repeat the first step to step 19 to investigate the error's impact under different rotation settings and report the rotation error E_R and the ADD error.

21. Based on the rotation error and ADD error reported, identify our expectation of the model's performance.

As one may have noticed, the error generation method we used in step 13 is not a conventional method where errors are extracted from a Gaussian distribution or a Uniform distribution. The consideration we had was that the probability of extracting an error from a Gaussian distribution or a Uniform distribution is uncertain. For instance, if one is trying to change the standard deviation of the Gaussian distribution from σ to σ' with $\sigma < \sigma'$ to achieve the increment effect of the distance between the 2D ground truth L_{2d} and the shifted landmarks L_{2d}^{err}, there is a chance that all the errors extracted from the distribution with σ' also falls into the distribution with σ, which makes the correlation between the standard deviation and the rotation error E_r and ADD error unclear. One may debate that the distance may serve as the independent variable, yet we do not have a guarantee that the errors extracted will not overlap with each other. The same argument applies to the decision not to use Uniform distributions. If one is wondering whether using Gaussian distribution with fixed standard deviation but moving the location of the mean to generate errors would be a choice, our opinion here is that having such errors is roughly the same as moving the 2D landmarks along the $y = x$ line in the Cartesian coordinate system, which has a great chance of resulting in error vectors all pointing to approximately the same direction, the randomness of the errors might be compromised.

Another question one may wonder is that why we decided to use the decomposition of the camera projection matrix instead of the translation and rotation vectors provided by the PnP algorithm as the ground truth. This decision was made based on the characteristics of the PnP algorithm. It has been proven that the accuracy of the PnP algorithm and the number of 2D-3D matching point pairs are positively correlated (Acuna and Willert, 2018), which means that there is an inherent uncertainty within the solutions given by the PnP algorithm, thus, using the resulting rotation and the translation from the PnP algorithm as ground truth would be a suboptimal movement.



(a) DRR in 2D

(b) DRR in 3D

Figure 4.5: 2D and 3D illustrations of the DRR [3]

4.4 Data Generation

Due to the reason that the attachment piece is newly designed, we have no existing real X-ray images at our disposal. In order to provide the training image for the neural network, the Digitally Reconstructed Radiograph (DRR) was used to simulate the behaviour of C-arms to generate artificial X-ray images. Furthermore, radiation exposure was another concern we had when making the decision. High doses of radiation have the potential to damage the DNA and lead to cancer [2]. However, massive data is preferred and needed to train the neural network. Therefore, to be able to obtain adequate amounts of images for the neural network and minimize unnecessary radiation exposure to the human at the same time, using DRRs as a substitute for real X-ray images is a cost-friendly, ethical and rational solution.

4.4.1 Digitally Reconstructed Radiograph

The digitally reconstructed radiograph is considered to approximate the authentic radiograph (Dorgham et al., 2012). After getting a 3D representation of the volume, the virtual X-ray beams are shot from the virtual source passing through the representation. Information created by the intersection of the X-ray beams and the representation is used to calculate the pixel intensity for the DRR. Such processes in 2D and 3D views are illustrated in Figure 4.5.

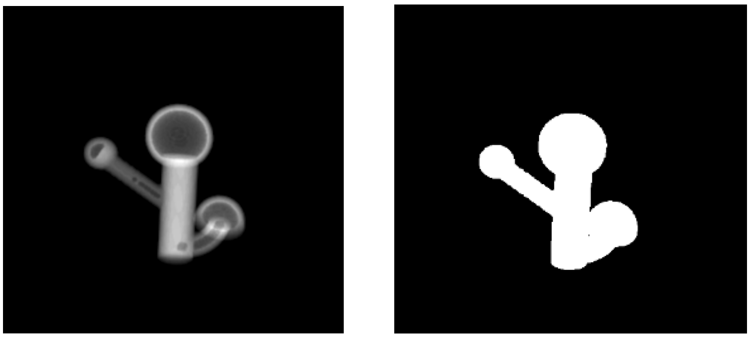
What needs to be remarked is that the rendering of the data volume used the Direct Volume Rendering technique, which preserves the gaseous phenomena and obeys the law of physics (emission, absorption, scattering)[4], due to the designing process of the attachment piece, it only has an outer shell, and it is hollow inside, the resulting DRR would resemble Figure 4.6(a).

Considering the instability issue that air may introduce to the attachment piece's radiograph, a solution we currently have is to fabricate the attachment piece in metal. The high radiopacity

[2] Source: https://www.cdc.gov/nceh/radiation/health.html

[3] Adapted from Montúfar et al. (2018)

[4] https://cgl.ethz.ch/teaching/former/scivis_07/Notes/stuff/StuttgartCourse/VIS-Modules-06-Direct_Volume_Rendering.pdf



(a) DRR of the current 3D representation of attachment piece

(b) mDRR of the attachment piece when it is solid and produced in metal

Figure 4.6: DRR and mDRR of the attachment piece

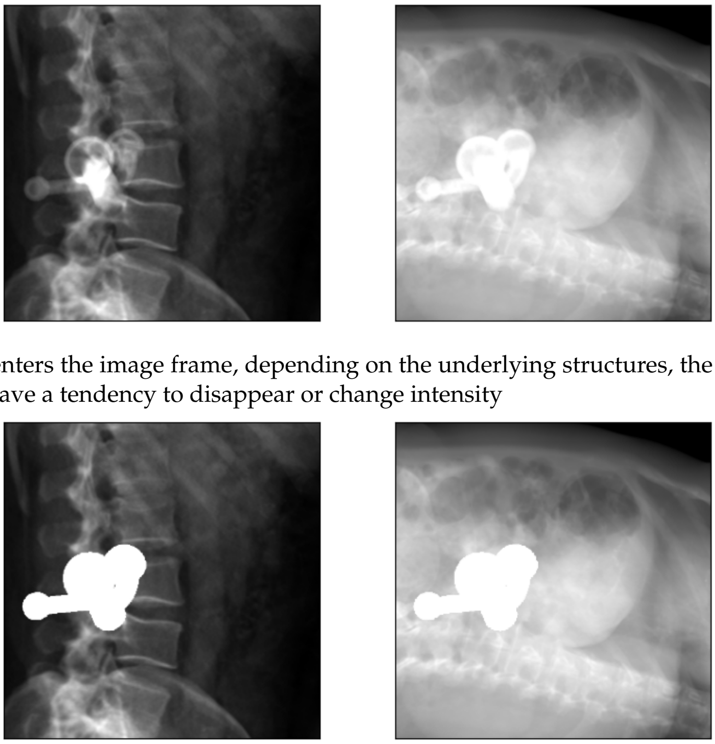
of metal ensures that the metal objects are able to attenuate the radiation when X-ray beams pass through them, which can block the visibility of any structures beneath them[5] and leads to regions with solid white-filled shapes of the outline of the metal objects, which consequently minimize the impact of air on tool's visibility in X-ray images. Based on this fact, we adjust our DRRs accordingly. Instead of using images similar to Figure 4.6(a) directly, we treated them as intermediate DRRs and extracted masks from them, which produced images that resemble 4.6(b) and then we used the masks as our final DRR (which will be referred as mDRR).

Intuitively, the DRRs of a hollow structure would be richer in the information it contains, and the neural network is able to benefit from it. This statement may be true when the DRRs contain only the attachment piece as in Figure 4.6(a). Yet, when human anatomy enters the image and overlaps with the attachment piece, the information that the hollow structure provided may be destroyed. For instance, overlapping the hollow structure with a bone has a chance to result in different pixel intensities in X-ray images compared to overlapping the hollow structure with soft tissues. And in most cases, the anatomy contains both bones and soft tissues, the pixel intensity would be more complex, as illustrated in Figure 4.7(a). However, when the attachment piece is manufactured as an opaque object in metal, its appearance in X-ray images may be less sensitive to the underlying structure, as shown in Figure 4.7(b). A trade-off needs to be balanced between the information a hollow structure may contain and the consistency of its appearance in X-ray images. This forms one of the reasons for our initial decision on the material of the attachment piece.

4.4.2 Data Acquisition

In order to preserve some degrees of reality, besides the mDRR of the attachment piece, the X-ray images or DRRs of human anatomy are also required to simulate real surgical environments where the attachment piece and the patient coexist. Fortunately, the DRRs of the anatomies are generated beforehand and provided. Holding the DRR as a tool, we form our dataset generation pipeline as follows:

[5] https://www.thoughtco.com/x-rays-and-metal-interference-608418



(a) When anatomy enters the image frame, depending on the underlying structures, the hollow structure in X-ray images may have a tendency to disappear or change intensity

(b) When anatomy enters the image frame, change of underlying structure does not affect the appearance of an opaque structure

Figure 4.7: Different appearances of the attachment piece in the X-ray images when the attachment piece is fabricated differently

Image Generation

1. Set rotation ranges and translation ranges for the rotation around X, Y and Z-axis, respectively.

2. Draw random values from the set ranges for rotation and translation.

3. Calculate transformation matrix A' based on the drawn rotation and translation using equation 4.1 to 4.6

4. Apply the transformation matrix A' inversely to the 3D volume and keep the virtual C-arm (virtual X-ray Source) untouched.

5. Generate intermediate DRR of the attachment piece under this transformation setting.

6. Extract the final mDRR from the intermediate DRR.

7. Superimpose the anatomy with the final mDRR.

Label Generation

The label consists of 3 parts, segmentation mask, 2D coordinates of the landmarks on the mDRRs and Gaussian heatmaps.

8. The segmentation mask is already generated along with the training image, it is the same as the final mDRR from Step 7

9. The 2D coordinates L'_{2d} are computed by plugging in the transformation matrix A' from step 3 and the 3D coordinates of the landmarks L_{3d} into Equations from (4.7) to (4.10)

10. The Gaussian heatmaps are calculated based on Equation (4.21), it is a slightly modified Gaussian. The denominator was removed from the original Gaussian distribution to prevent the value from approaching 0 when the standard deviation increased. The two variables, x and y are not only the 2D coordinates of a landmark but also the means of the modified Gaussian distribution, and s_g is a constant to scale up the Gaussian.

    Then a crop operation is performed to set the background pixel value of the heatmap to 0. By doing so, we intended to have a clear boundary between the important region and the less important region.

    Each landmark has exactly one heatmap generated around its 2D coordinates. In total, 6 heatmaps need to be generated for one image and stacked on top of each other, which means the final Gaussian heatmap for a single image has 6 channels.

$$h(x, y) = s_g e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.21}$$

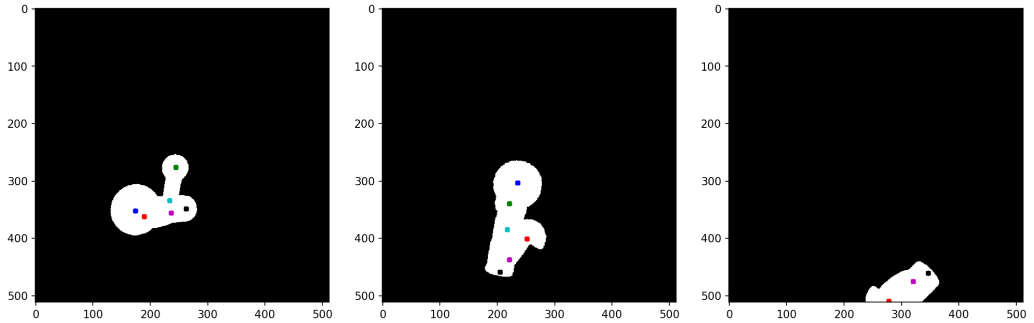
11. Loop through this process from step 2 to step 10 to populate the dataset until it reaches a satisfying size.

4.4.3 Attachment Piece 2D Pose Filtering

As one may notice, when generating mDRRs of the attachment piece, the rotation and the translation applied to the 3D volume are randomly drawn from defined ranges. Particularly, when the span of the specified ranges get large, the resulting 2D pose in the mDRRs of the attachment piece might be challenging for human to mentally reconstruct its 3D pose as in Figure 4.8a. Or some of the 3D landmarks are blocked by the attachment piece itself when casting the X-ray and results in blocking partially its own structures as in Figure 4.8b. There is also a chance for more than 2 points to be located outside the image frame as in Figure 4.8c, which will lead to the failure of the PnP algorithm as mentioned in 4.2 to find a unique pose PnP algorithm needs at least 4 points. Additionally, our presumption is that it might be easier for the neural network to learn the landmarks' location when the attachment piece's three-branch structure is well maintained in the 2D images. To be able to separate these kinds of images (will be referred to as challenging images) from the ones with all landmarks within the image frame and three branches distinguishable (will be referred to as unchallenging images) as in Figure 4.6b, we proposed several criteria based on the characteristic of the 2D pose. And two widgets are also developed in case of one does not have confidence in the criteria. By using the widgets, one would be able to categorize the images into unchallenging data or challenging data manually.

Criteria for Automatic Filtering

This section introduces the 3 criteria set for filtering unchallenging and challenging 2D poses.



(a) Strange 2D Pose (b) Landmarks partially blocked (c) More than 2 landmarks outside the frame

Figure 4.8: mDRRs with hard-to-interpret 2D pose of the attachment piece

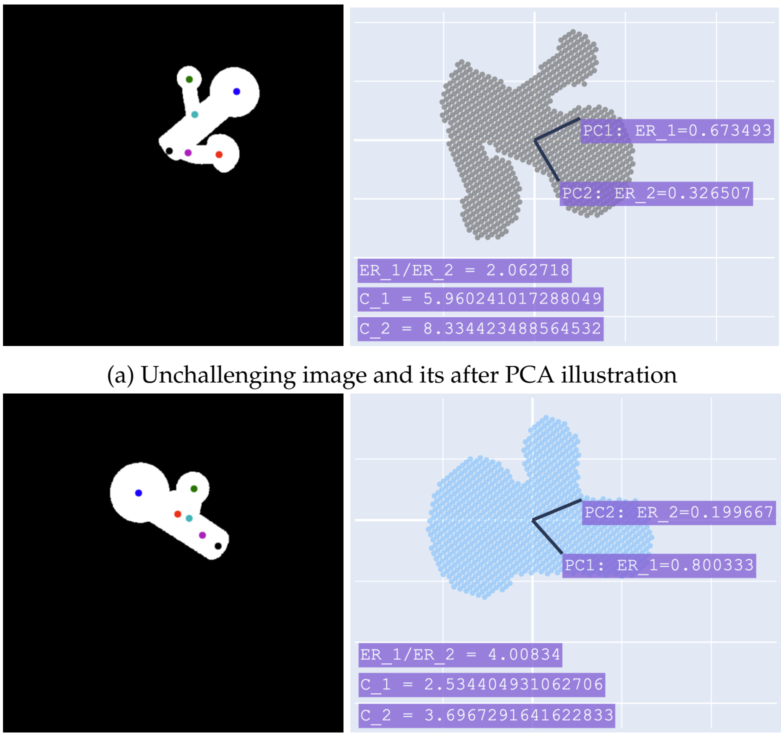
Criterion 1 By observing the unchallenging and challenging images, a pattern of the distribution of the pixels coloured in white can be found. The white pixels are more spread in the unchallenging images than in the challenging images due to the distance of the branches from each other. Which inspired us to treat the white pixels as data points instead of images. The locations of the white pixels can be extracted to form a new dataset with 2-dimensional records, and then use the PCA algorithm to explain the variance in two directions. After feeding the generated dataset to PCA, the explained variance will be returned, representing the amount of variance explained by each principal component. Figure 4.9 is an example of the described process. In Figure 4.9(a) and Figure 4.9(b), the after-PCA transformed data points (centred at 0), the direction of two principal components and the explained variance ratio of them ER_1 and ER_2 are displayed on the right side. From the intuitive value of the explained variance ratio, a preliminary conclusion can be made that under the unchallenging image cases, the contribution of ER_1 and ER_2 to the distribution are more close to each other than under the challenging image cases, which leads to the fractions $\frac{ER_1}{ER_2}$ smaller than the ones under the challenging image cases.

However, there is no absolute guarantee that the above observation holds true for all the cases, but when the attachment piece is in a challenging pose, the landmarks appear to be more clustered than they are in an image with an unchallenging pose. This motivated us to consider incorporating the sum of the landmarks' pairwise distances. Therefore, Criterion 1 is formed, which can be seen in Equation (4.22). The $dis_{i,j}$ stands for Euclidean distance between landmark i and j. The s_1 value on the right-hand side of the inequality is an empirical value we gained from observation.

$$C_1 = \frac{\sum_{i=1}^{6}\sum_{j=1}^{6} dis_{i,j}}{100 \cdot \frac{ER_1}{ER_2}} > s_1 \tag{4.22}$$

Criterion 2 Criterion 2 is an extension of Criterion 1. We further stressed the importance of pairwise distances by adding the length of the skeleton. The skeleton is marked out with a light yellow dash line in Figure 4.10, which holds the shape of the 2D attachment piece we intend to keep. The landmarks are numbered from top to bottom, left to right, and from 1 to 6, as shown in Figure 4.10. The resulting inequality is shown in Equation (4.23). Same as s_1, s_2 is also an observed value. This criterion can be used either separately or jointly with Criterion 1.

$$C_2 = \frac{\sum_{i=1}^{6}\sum_{j=i}^{6} dis_{i,j} + (dis_{1,2} + dis_{2,6} + dis_{1,6} + dis_{3,6} + dis_{1,3})}{100 \cdot \frac{ER_1}{ER_2}} \geq s_2 \tag{4.23}$$



(a) Unchallenging image and its after PCA illustration

(b) Challenging image and its after PCA illustration

Figure 4.9: Unchallenging image and challenging image before and after PCA illustration

Criterion 3 Criterion 3 is a rather strict criterion, it forces a pairwise distance value to be in a certain range for an image to be considered an unchallenging image. It consists of 6 inequalities, as shown in Equation (4.24). In the first 4 inequalities, $\widehat{dis}_{i,j}$ where $1 \leq i \leq 6$ and $1 \leq j \leq 6$ stands for the Euclidean distance between landmark i and landmark j in a 2D image with no 3D transformation (i.e, neither translation nor rotation is applied to the 3D volume of the attachment piece, and the C-arm is also in its initial position, which results in the front view of the attachment piece in 2D image). The last two inequalities restrict all the landmarks within the image frame. In our case, the height and the width of the DRR are the same, we represent them as l_{DRR}.

$$\begin{aligned} dis_{1,2} > \frac{1}{2}\widehat{dis}_{1,2} \quad &and \quad dis_{1,3} > \frac{1}{2}\widehat{dis}_{1,3} \\ dis_{1,6} > \frac{1}{2}\widehat{dis}_{1,4} \quad &and \quad dis_{3,5} > \frac{1}{2}\widehat{dis}_{3,5} \\ 0 < x_i < l_{DRR} \qquad &\quad 1 \leq i \leq 6 \\ 0 < y_k < l_{DRR} \qquad &\quad 1 \leq j \leq 6 \end{aligned} \tag{4.24}$$

Widgets Supporting Data Acquisition

As the three criteria are relatively naive, they cannot guarantee drawing a clear line that perfectly separates the unchallenging and challenging images. There are edge cases that belong to unchallenging images but were categorized as challenging images. Yet, fortunately, the s_1 and s_2 values we picked combined with the third criterion were able to make sure that no challenging image

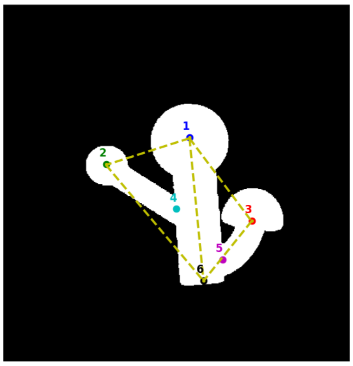


Figure 4.10: Skeleton of the attachment piece

was categorized as unchallenging. The workflows of the widgets are slightly different and will be presented in detail in the following.

Widget 1 In order to tackle the misclassified unchallenging cases, widget 1 was developed. The widget is able to ask for human opinion when the value of criterion 1 C_1 or criterion 2 C_2 falls into a predefined margin range where $s'_1 \leq C_1 \leq s''_1$ or $s'_2 \leq C_2 \leq s''_2$. Then the image is categorized based on the human's decision.

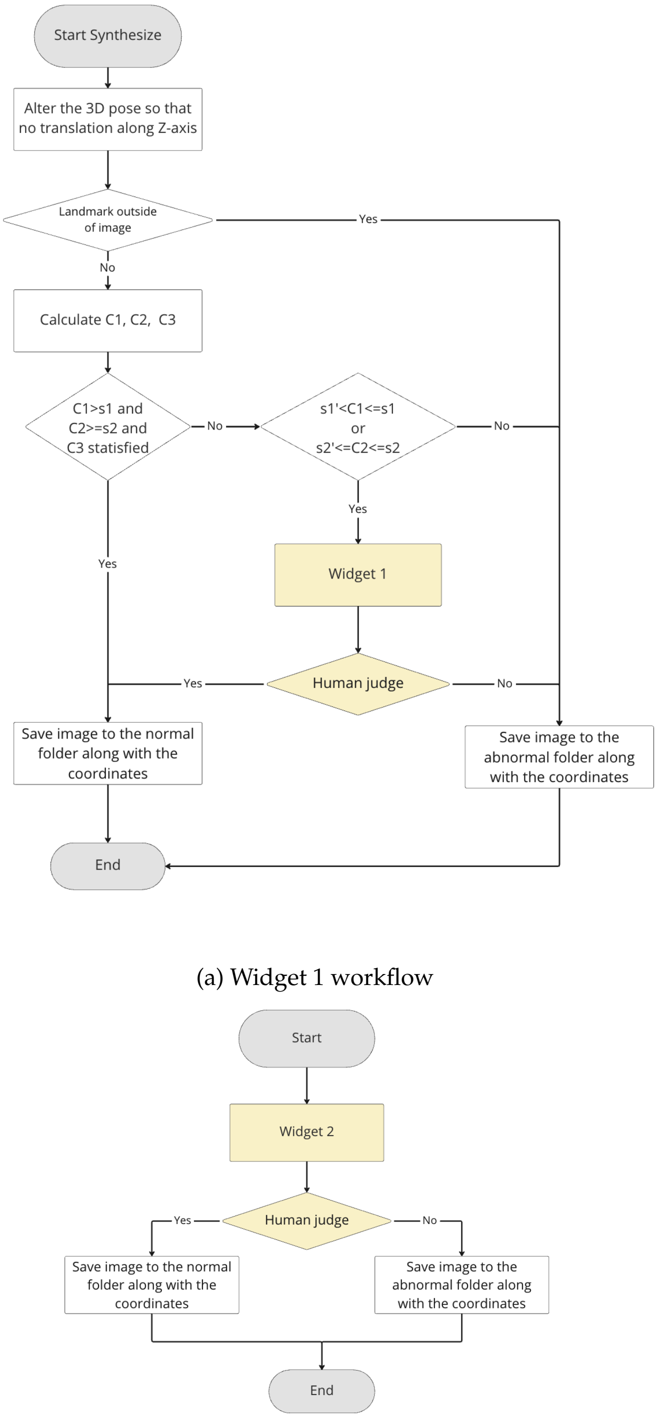
The workflow of Widget 1 can be seen in Figure 5.1(a). After starting the mDRRs synthesize process, it is important to record the rotation vector and the translation along X-axis and Y-axis, they are needed to generate a replica that has the same transformation except for the translation along Z-axis. Before the calculation of C_1 and C_2, if any of the landmarks is outside the image, then this image will be categorized as a challenging image immediately. Then the C_1, C_2 and the first 4 inequalities in Criterion 3 (which will be referred to as C_3 for simplicity) are calculated based on the replica image. If an image survives the criteria with its $C_1 > 4$, $C_2 \geq 8$, and its landmark pairwise distances satisfy Criterion 3, then this image will be categorized as an unchallenging image automatically. But if it cannot pass one or multiple criteria, the margin range will be used to perform another filtering. Note that the s'_1, s''_1 values and s'_2, s''_2 values are purely dependent on the user's preference for the strictness of the filter. If any of the C_1 and C_2 values fall into the margin range, the human's opinion will be asked. But if both C_1 and C_2 values cannot fit in the margin range, then it will be considered as a challenging image.

Widget 2 We also developed a widget in case one intends to categorize all the generated images manually.

By using widget 2, the responsibility of categorizing images entirely falls on the shoulders of the user, which means it has a naive logic as shown in Figure 5.1(b). After starting to run the widget, the image is generated per the user's command. The images are also saved to corresponding folders based on the user's decisions.

4.5 Model Design and Training

As mentioned in Chapter 3, the prerequisites for using the PnP algorithm are the camera's intrinsic parameters, 3D coordinates of points on the object, and their corresponding 2D coordinates. Under our project settings, the camera's intrinsic parameters and 3D coordinates of the object are


Start Synthesize
Alter the 3D pose so that no translation along Z-axis
Landmark outside of image
Yes
No
Calculate C1, C2, C3
C1>s1 and C2>=s2 and C3 statisfied
No
s1'<C1<=s1 or s2'<=C2<=s2
No
Yes
Yes
Widget 1
Yes
Human judge
No
Save image to the normal folder along with the coordinates
Save image to the abnormal folder along with the coordinates
End
Start
Widget 2
Yes
Human judge
No
Save image to the normal folder along with the coordinates
Save image to the abnormal folder along with the coordinates
End

(a) Widget 1 workflow

(b) Widget 2 workflow

Figure 4.11: Widget 1 and widget 2 workflows

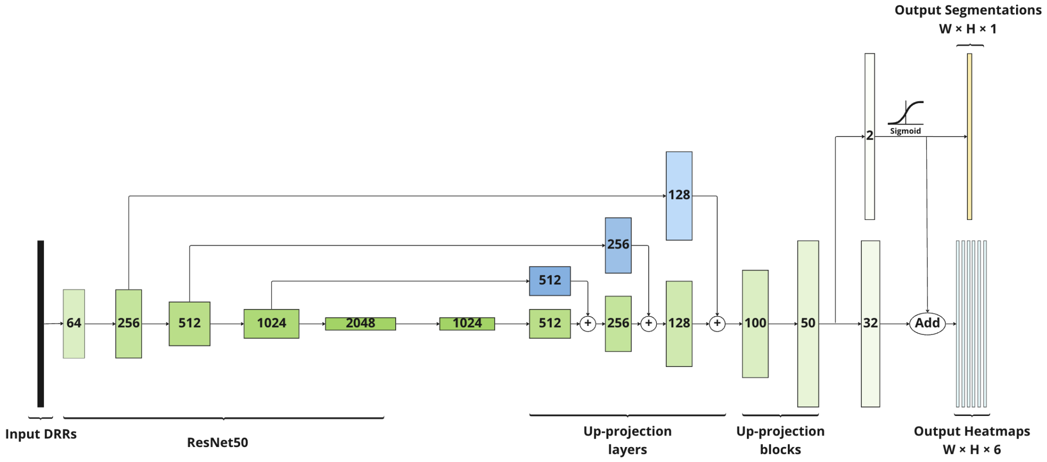


Figure 4.12: Overview Architecture

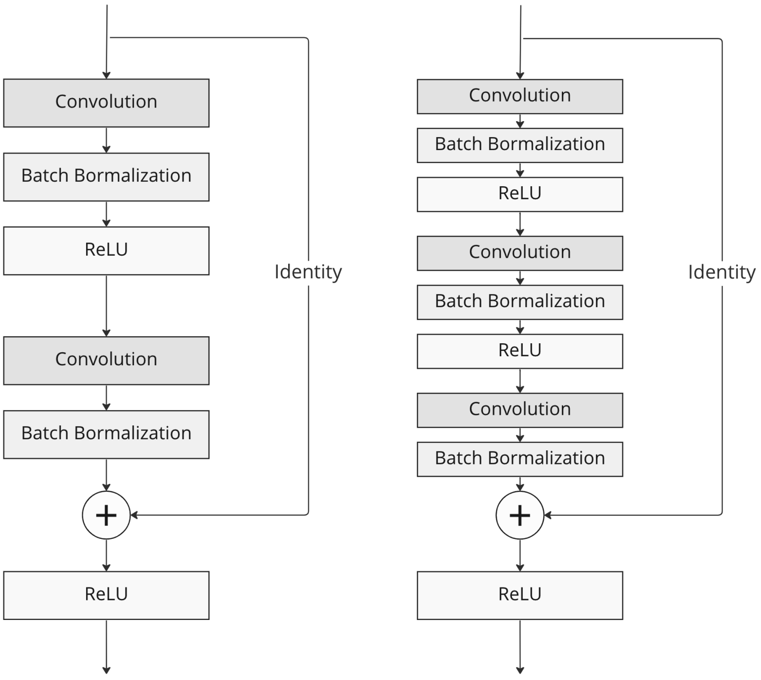
at our disposal; combined with the 2D coordinates produced by the landmark localization step, the camera's/object's pose can be relatively easily determined, which is one of the reasons why we are focusing on achieving the landmark detection task based on the 2-steps approach. In addition, due to the fact that depth information is absent in X-ray images, we are limited to using only greyscale cues.

4.5.1 Model Architecture

We based our work on the CSL model proposed by Laina et al. (2017). The network was constructed with an encoder block and a decoder block and has two output branches for the segmentation task and localization task separately. The overview of the architecture we propose can be seen in Figure 4.12. In the remainder of this section, the model's architecture will be presented in detail.

Encoder

For the encoder, we also took advantage of the ResNet50 (He et al., 2016a) as in the original CSL model. As pointed out in Section 3.3, the residual connection is capable of saving neural networks from struggling with performance issues when the structures become deeper and more complex. The realizations of such connections are the residual blocks, consisting of layers of convolution operations and a residual connection from block input to block output. Two types of composition for the residual blocks have been developed as well, one has only two layers of convolution between the input and output, and the other one has 3. There are two factors that ResNet50 are preferred over all the variants of ResNet, the first one is performance. ResNet50 uses the 3-layer residual block as in Figure 4.13b, and it achieved higher accuracy on ImageNet compared to RetNet18 and ResNet34, which use 2-layers residual blocks as in Figure 4.13a (He et al., 2016a), the second one is the computation time, the deeper a network gets, the more computation power and time it requires when training, we attempted to find a balance that can fit our limited computation resources while preserving the model's ability and performance as much as possible.



(a) Two-layer residual block (b) Three-layer residual block

Figure 4.13: Two types of residual blocks used in ResNet

Decoder

The decoder part of the network was designed based on the CSL model. It composes of several up-sampling blocks and skip connections from the encoder, which manage to pass low-level but high-resolution information from encoder to decoder. We also adopted the up-projection block (see Figure 4.14) introduced in Laina et al. (2016) to achieve up-sampling. The up-projection block exploits the residual connection from the ResNet, the input first goes through an unpooling layer which expands the resolution of the feature maps. After that, the enlarged feature maps are fed to two convolutional layers with 5 × 5 kernel separately. One serves as the shortcut connection, the other one is fed into another convolutional layer with 3 × 3 kernel, and the outputs of the shortcut connection and the 3 × 3 convolution then be summed together and fed into a ReLU layer. It is worth noting that for the unpooling operation, instead of using max-unpooling which expands the feature maps based on the location of max values recorded during pooling, we adopted the regular-grid unpooling as in Figure Dosovitskiy et al. (2015). As shown in 4.14, the unpooling layer increases the feature map to twice the original size, with the max values put to the left corner of each 2 × 2 block and leaving the rest blank. As for the skip connections, we first extracted the intermediate layer's output from ResNet and fed it into a convolutional layer with kernel size 3 × 3, then the output of the convolutional layer is directly combined with the output of the up-projection block using pairwise summation to form an up-projection layer. In total, 3 up-projection layers are stacked together. Following this, we added two additional up-projection blocks to be able to have the final output size the same size as the input image size. Afterwards, the model is split into two branches, with one branch concentrating on the segmentation task and the other one focusing on extracting features for the heatmap. Afterwards, the segmentation branch's output is added to the localization branch. This decision intends to

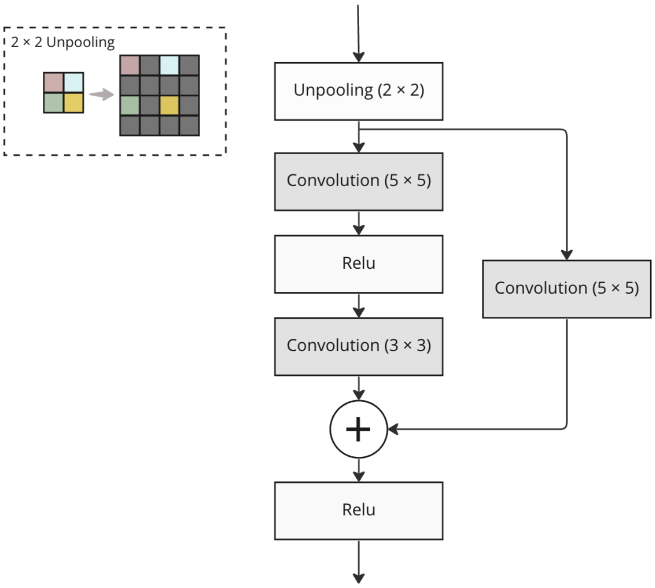


Figure 4.14: Up-projection block

express the importance of an area in the feature map where the attachment piece is predicted to be while leaving the rest of the area unaffected. In the end, the final heatmaps are produced with a 1×1 convolution. As one can notice, the entire model follows the fully convolutional fashion, which has been shown to be more suitable and preferable when dealing with image-related tasks (Long et al., 2015).

4.5.2 Loss Function

Loss for Segmentation Branch

For the segmentation branch, we use one channel to express the background and foreground, as 0 represents the background and 1 represents the attachment piece region. We used Dice Loss (Sudre et al., 2017) as the loss function for this branch. It is a variant of the Dice Score Coefficient (DSC). The DSC is used as a metric in computer vision to measure the similarity between two images (Jadon, 2020). It has been adapted to become an objective function as Dice Loss. The 2-class Dice Loss can be expressed as Equation (4.25). The ϵ in numerator and denominator is used to ensure numerical stability, which assures the function does not have 0 as the denominator when prediction and ground truth are both 0.

$$DL_2 = 1 - \frac{2\sum_{i=1}^{w}\sum_{j=1}^{h} S_{i,j}\hat{S}_{i,j} + \epsilon}{\sum_{i=1}^{w}\sum_{j=1}^{h}(S_{i,j} + \hat{S}_{i,j}) + \epsilon} \tag{4.25}$$

Loss for Heatmap Regression Branch

The mean squared loss is adopted for the heatmap regression branch. Equation (4.26) is the formula for calculating the loss. w and h represent the image's width and height, while n represents the number of heatmaps (landmarks).

Therefore, the overall loss for the model training is given by equation (4.27), where λ_{ce} and λ_{MSE} are contribution coefficients.

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\sum_{x=1}^{w}\sum_{y=1}^{h}\|y_{x,y,i} - \tilde{y}_{z,y,i}\|_2^2 \tag{4.26}$$

$$L = \lambda_{DL_2}DL_2 + \lambda_{MSE}L_{MSE} \tag{4.27}$$

4.6 Post Processing

When the model reaches the inference stage, the ideal prediction for each landmark would be a perfect heatmap with exactly one peak. Yet that is not always achievable in reality, nevertheless, our expectation for the model is to perfectly capture the region where the landmarks should lay instead of a complete matching of the pixels' response value. Thus a post-process procedure is needed to extract the landmark's coordinates from the predicted heatmap. The initial proposal was a rather simple one. The landmarks' locations are generated by finding the pixel's location that has the highest response value. Although it is intuitively sound, we can neither guarantee that the pixel with the highest response is right in the centre of the predicted heatmap nor that there is only one pixel with the highest response. For this reason, we need to find another approach to locating landmarks.

In our current using procedure for post-processing, the heatmaps need to go through three steps. The first step is to create a mask of the predicted heatmap based on a threshold. The second step is to detect a blob or circle in the mask, which ideally centres around the centre of the response area. The second step was achieved with the help of OpenCV[6], detailed functions used for this step will be introduced in Chapter 5. And at the final step, we locate the centre coordinates of the detected circle.

[6] https://opencv.org/

Chapter 5

Experiments and Results

5.1 Implementation details

5.1.1 Model performance Expectation Generation

Sensitive Poses Finding

Following the process defined in section 4.3, we first did a systematic error analysis and tried to find the 3D rotation of the attachment piece under which the PnP algorithm is most sensitive to the error. We believe by doing so, we can use this 3D rotation to find a rather strict error tolerance range.

We set the rotation around the X and Y-axis from $-90°$ to $90°$ and omitted the rotation around the Z-axis. Our reason is that the Z-axis is perpendicular to the image plane and pointing outwards, the rotation around Z-axis is the last to be applied, which results in a rotation in 2D. Hence, we assume that the rotation around this axis has less impact on the stability of the PnP algorithm than the rotation around the X-axis and Y-axis in 3D. The translations were also left out. The process was put in a nested loop, and the interval was set to $10°$. We investigated the influence of the magnitude of the error vector $\hat{ei}$ on the rotation and 3D reconstruction by applying the error E_f to all 6 landmarks with gradually increased magnitude from 0 to $10mm$ (under this project setting, 1 pixel equals to $1mm$). For each error vector magnitude, we drew 300 error samples randomly and added them to the ground truth 2D coordinates. As a reminder, the direction of each error vector $\hat{e_i}, 1 \leq i \leq 6$ is random, only the magnitude of the error vectors was the same.

As stated before, this error simulation procedure depends on the PnP algorithm to find the rotation and the translation, for which we used the $cv2.solvePnP$ function provided by OpenCV[1] and set the $flag$ to $cv2.SOLVEPNP\_EPNP$. The $cv2.solvePnP$ function returns a rotation vector r_{pnp} in radians and a translation vector t_{pnp}. In order to calculate the rotation error E_R using quaternion, we used $cv2.Rodrigues$ also provided by OpenCV to transform the rotation vector r_{pnp} to a rotation matrix R_{pnp}, then we used the $spatial.transform.Rotation$ from SciPy [2] to transform the rotation matrix R_{pnp} to its quaternion form. The rotation matrix R_{qr} returned by QR decomposition from Scipy[3] was also transformed to quaternion using SciPy. After finishing calculating the rotation error E_R and the ADD error, we recorded and reported their median. The poses that generated the highest median rotation error and the pose that generated the highest median ADD error are identified as the most sensitive poses. The purpose of trying to find

[1] https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.transform.Rotation.html

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.qr.html

sensitive poses is to identify an upper bound for the 2D error, based on which our initial expectation (upper bound) for the model's performance about the median average Euclidean distance between the prediction and ground truth is formed. The hypothesis is that the performance expectation formed under sensitive pose settings should be a rather strict bound for other poses, and the PnP algorithm should have more tolerance to errors and perform better under other pose settings. This range can be used not only to assess the model's performance in the current project but also as a reference for other landmark-locating methods that will be explored for X23D in the future.

Zoom-in on Sensitive Poses

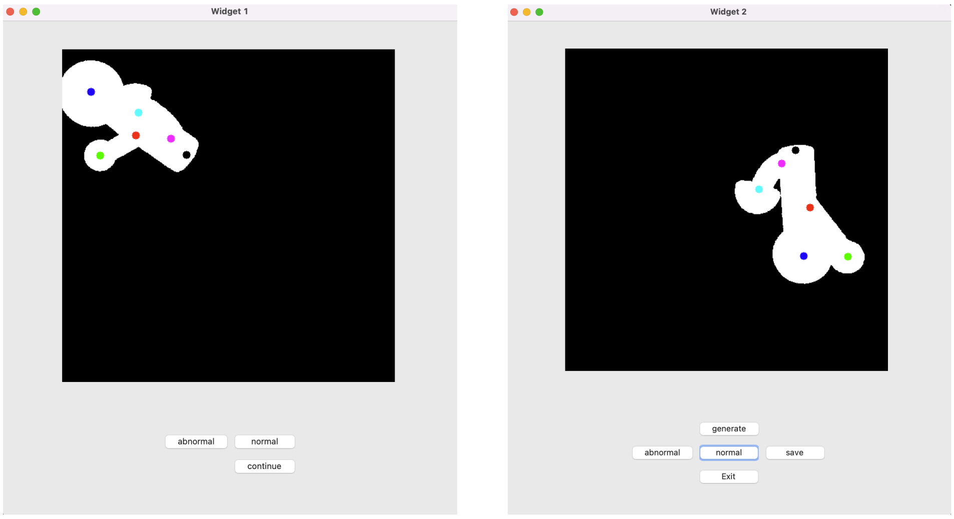
After the sensitive poses had been identified, we performed a deeper inspection of how errors impact the accuracy of the PNP algorithm under these poses. Unlike in the previous step, where the errors were applied to all the landmarks at once, in this step, errors are added to the landmarks one at a time, from only adding an error to one of the landmarks to adding errors to all the landmarks (the landmark with error added will be referred as a compromised point). And we investigated the relationship between the magnitude of the error vectors and the accuracy of the PnP algorithm every time we added a new compromised landmark. We again used a nested loop for this. After setting the rotation of the 3D volume based on the detected sensitive poses. We then set the range for error vectors' magnitude to $[0, 10]$ with 1 as the interval and then set the number of compromised points range from 1 to 6. In the next step, we looped over the magnitude's range and drew 300 samples for each magnitude. Furthermore, due to the fact that we are operating in the medical domain, we prefer the error to be as small as possible. Therefore, we zoomed in on the magnitude of the error vector by setting the magnitude to change from 0 to 3 with 0.3 as the interval, and we repeated the above procedure, and the results were recorded and reported.

5.1.2 Criteria Lower Bounds Identification for Unchallenging Images

Lower Bound Generation for Criterion 1

As mentioned in section 4.4.3, the lower bound value s_1 of criterion 1 is an observed value. To obtain a value for s_1, we investigated the value of C_1 by rotating the attachment piece in 3D along X-axis and Y-axis individually from $-90°$ to $90°$. The rotation along Z-axis is omitted as before. However, the existence of the degree of freedom for the translation along the Z-axis when moving the attachment piece in 3D leads to a zoom-in and zoom-out effect and makes the size of the attachment piece in the mDRRs changeable. This phenomenon has some degree of impact on the value of $\frac{ER_1}{ER_2}$ and the distance sum. For the sake of simplicity, the translations along all three axes are set to 0. After, we inspected the result with the rotation around X-axis and Y-axis as independent variables and C_1 value as a dependent variable. Meanwhile, to form an initial impression of how the C_1 values are different under the challenging cases and unchallenging cases, we calculated the C_1 for images that contain several slightly different lateral views and top-down views for the challenging images and for the unchallenging images we calculated the C_1 value for images that contain the front views of the attachment piece. Then we compared our impression against the results and manually checked the corresponding poses with C_1 that lies around our impression boundary to identify the lower bound.

We identified 4 as our value for s_1. As an example, the poses in Figure 4.9 can be well separated with $s_1 = 4$. It should be emphasized that this exploration process did not involve any translations, which means when using $s_1 = 4$ with random rotation setting and translation setting, no matter how much the translation was set for the movement along Z-axis, the 3D volume



(a) Widget 1's user interface

(b) Widget 2's user interface

Figure 5.1: Widgets' user interfaces

needs to be pushed backwards or bring forward to the plane where the translation along Z-axis equals to 0 while keeping the rotation around all axes and the translation along X and Y-axis intact to calculate the value for the comparison.

Lower Bound Generation for Criterion 2

The investigation process is the same as how s_1 was generated, but with different dependent variable calculation formulas. Manual checking is also required to identify the value for s_2.

We identified 8 for s_2. The poses in Figure 4.9 are also an example as they can be well separated with $s_2 = 8$. When using this criterion to filter the images, the same Z-axis translation alternating process stressed in Criterion 1 also needs to be employed.

5.1.3 Data Acquisition Supplemental Widgets Development

For the development of the widgets, considering python is our chosen language to implement the neural network, to have consistency, we also used python to build the widgets. For the graphical user interface implementation, the tkinter[4] package was used.

Widget 1 User Interface

Figure 5.1a shows the user interface of widget 1. When the C_1 or C_2 hits the set range, the widget shows up with the current confusing image with landmarks' locations indicated on it, humans can categorize the current image by clicking the button below the image. After the human makes the decision, "continue" button can be clicked to resume the image generation process.

[4] https://docs.python.org/3/library/tkinter.html

Widget 2 User Interface

The user interface of widget 2 is shown in Figure 5.1b. After initiating widget 2, an image starts the generation pipeline after the user clicks on the "generate" button. As soon as the image is ready, the widget is updated with the newly generated image, which also has landmarks' locations indicated on it. Afterwards, the user is able to provide the verdict based on which the image is categorized. The user can always suspend the process by clicking on the "Exit" button.

5.1.4 Dataset Generation

As we mentioned before, we were using the DRR to generate simulated X-ray images of the attachment piece. Thanks to Dr Hooman Esfandiari, the code for producing DRRs and mDRRs are already provided. Using the code and our criteria, 14000 unchallenging images and 80000 challenging images were created. As for the anatomy DRRs, Dr. Hooman Esfandiari can also take the credit, he provided around 40000 anatomy DRRs which facilitated the dataset generation process. The remaining work is to overlay the anatomy DRRs with the mDRRs of the attachment piece. An additional step we did before the overlay was to resize the mDRRs with the size 512 × 512 to have the same size as anatomy DRRs which were in size 320 × 320. The $resize$ function from OpenCV was used to downsampling the mDRRs. After resizing the images, the overlaying was carried out by using the function $cv2.addWeighted$ provided by OpenCV. To refresh the memory, our preliminary decision is to fabricate the attachment piece in metal, so we set the weight for both the mDRRs of the attachment piece and the DRRs of anatomy to 1 when performing the overlay. We also added data augmentations to the mDRRs of the attachment piece and the DRRs of the anatomy using Albumentations[5] before the overlay operation according to a random number that controls the probability of applying the augmentations. The augmentation methods we chose from Albumentations, including $Flip$, $Transpose$, $Blur$, $RandomBrightnessContrast$, $Downscale$, $GaussNoise$, $MotionBlur$, and $PixelDropout$, and the augmentations were also applied to the 2D coordinates of the landmarks.

By combining the datasets on hand, the approximately 10000 anatomy DRRs, the first 4000 unchallenging images, and the last 30000 challenging images were used to generate two test sets and validation sets. One contains only the unchallenging images, and the other contains both unchallenging and challenging images. Each contains 6000 test images and 3000 validation images, and the remaining unused anatomy DRRs and attachment piece mDRRs will be used to generate training data during the training. The test and validation set that contains only the unchallenging pose of the attachment piece is referred to as the unchallenging dataset. And the other test set and validation set consists of 33% of unchallenging poses and 67% challenging poses of the attachment piece, and we refer to it as the general dataset. Figure 5.2 is an intuitive illustration of the mentioned process. Regarding the training data, we decided to perform the augmentation and the overlay on the fly during the training. The heatmaps for the training and validation were also generated while training, the standard deviation σ was set to 10 to have a bigger response area, which presumably would be easier for the model to learn. And the scalar s_g in Equation (4.21) was set to 30 to scale up the Gaussian. Then cropping operation extracted the centre of the heatmaps using a box with 2.5σ sidelength and set the rest region to 0.

5.1.5 Model Implementation

For the implementation of the neural network, we used TensorFlow[6]. The backbone network ResNet50 was also provided by TensorFlow. After loading the ResNet50, we first removed the last layers by extracting its intermediate results by the layer's name, and a "Conv2D" layer with

[5] https://albumentations.ai/

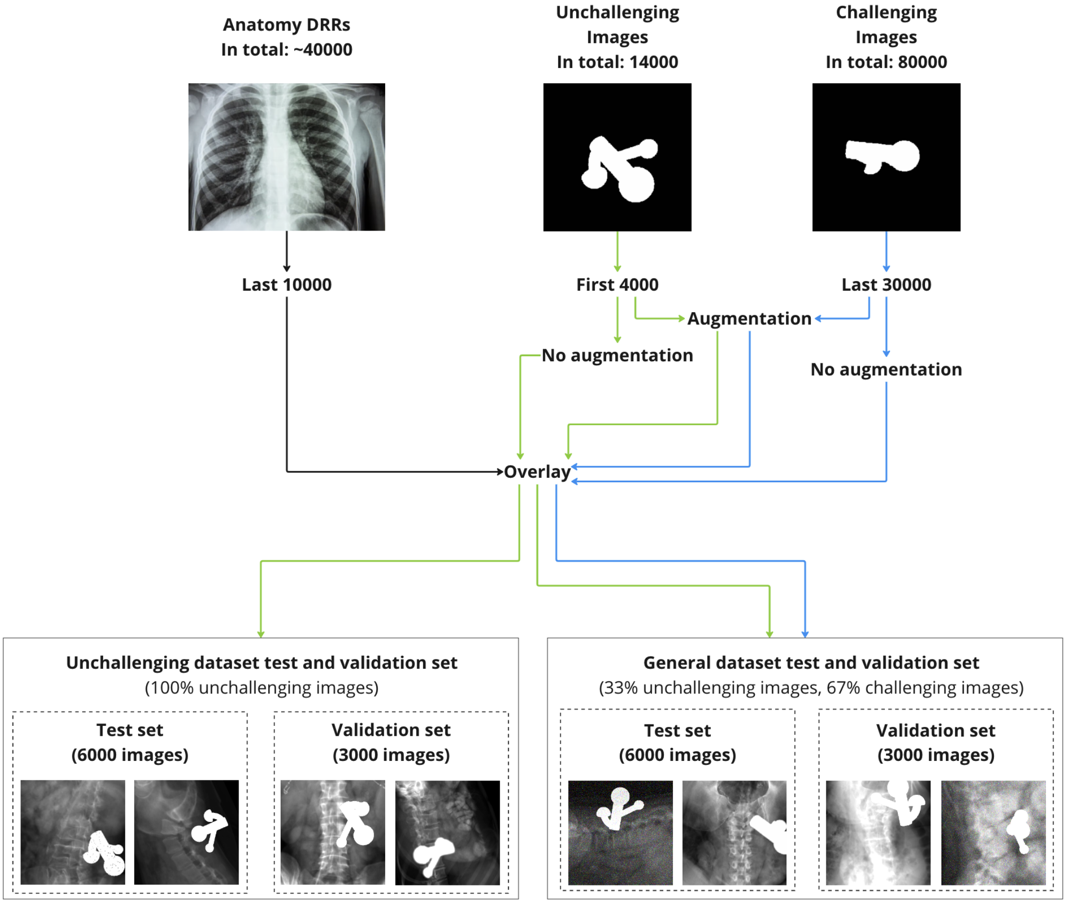


Figure 5.2: Structure of test and validation set

kernel size 3, "same" padding and the ReLU activation were used to process the output from ResNet50 before feeding into succeeding up-projection layers. To implement the up-projection layer in the decoder, we first created the up-projection block. For that, we first implemented the rigid unpooling function and transformed it into a layer using $tf.keras.layers.Lambda$, two $Conv2D$ layers using ReLU as activation function and "same" as padding style were subsequently added to the main branch with kernel size 5 and 3, respectively. And for the skip connection, a $Conv2D$ layer that has the same activation and padding style was added to process the output of the unpooling layer. Next, the main branch's output and the skip connection output were added together, and a final ReLU layer was used to process the result. For the skip connections from ResNet50, we took the output from layers with names "conv4_block6_add", "conv3_block4_add" and "conv2_block3_add", and these outputs were then passed to a $Conv2D$ layer with kernel size 3, "same" padding, and ReLU activation. And we added the outputs with the outputs from the up-projection blocks to form the up-projection layers. After the three consecutive up-projection layers, we reused two additional up-projection blocks to upsample the output to the original image size. From here, the model is split into two branches. A "Conv2D" layer with kernel size 1 and an activation layer with "sigmoid" activation was used for the segmentation branch. Another "Conv2D" layer with kernel size 3, ReLU activation and "same" padding was used before the adding operation of the segmentation branch and localization branch. At the end of the output layer of the localization branch, another "Conv2D" layer was used with ReLU activation and kernel size 1 to form a fully convolutional network. The intuition for using ReLU is that the maximum of the Gaussian exceeded 1 due to the scaler applied when generating heatmaps, hence, we intended not to set an upper bound for the model.

5.1.6 Model Training

Training Data On-the-fly Generation

The training data was generated using the same procedure as the generation for the test and validation sets. The first 30000 images from the anatomy dataset were used. When training on the unchallenging dataset, the last 10000 mDRRs from the unchallenging attachment piece alone dataset were used. The first 50000 mDRRs from the challenging attachment piece alone dataset are also included when training on the general dataset. The images were overlaid and augmented the same way as we did for the test and validation set. At each epoch, 60000 images were generated for the training on the unchallenging dataset, and 90000 images were generated for training on the general dataset. The difference in the number of training images is owing to the "challenging" we defined, the assumption is that by increasing the number of training data, the model might be able to learn the challenging cases better compared to lesser training data.

Model Training

The models were trained on both the general dataset and the unchallenging dataset with an Adam optimizer with a decay rate of 0.6 every epoch, the initial learning rate was set to 0.001. Early stopping was adopted to monitor the loss of the localization (heatmap regression) branch because of the importance of the performance of the heatmap branch to us. When the loss does not decrease more than 1 epoch, the training is automatically stopped, and the weights that achieved the best (lowest) loss on the validation set are stored by adding the callback function $ModelCheckpoint$ provided by Tensorflow.

5.1.7 Post-processing Implementation

The first step in post-processing is to create a mask from the predicted heatmap. We extracted all the pixels with values larger than 10 (one-third of the scalar s_g applied to the Gaussian), and then set these pixels to 1 and left the rest region with 0. To detect the circle in the mask, we used blob detection provided by OpenCV to find the blob in the mask. We set the parameter $filterByCircularity$, $filterByInertia$ and $filterByConvexity$ all true and $filterByColor$ false, and we set the value of $minCircularity$, $minInertiaRatio$ and $minConvexity$ all to 0.01. And the blob detection outputted the blob centre's coordinates, and we treated these coordinates as the final predicted coordinates of the landmark. However, there is a possibility that the blob detection may fail. In that case, we used the HoughCircles from OpenCV to detect a circle in the heatmap and output the centre coordinates of the detected circle. If no coordinate is given by HoughCircles, we consider that no landmarks were predicted. The order of these two centre-finding methods is important to us. The coordinates that HoughCircles provides are always integers or half-integers, whereas blob detection can give us more precise coordinates with a resolution of at least 0.00001. One may prefer integers or half-integers over high resolution, but under our project setting, high resolution is favoured.

[6] https://www.tensorflow.org/

5.2 Results

5.2.1 Performance Expectation

Sensitive Poses Identified

As promised in section 5.1.1, we report our result in Figure 5.3 and Figure 5.4. The X-axis represents the rotation around X-axis and Y-axis in degrees in the 3D world with respect to the attachment piece's coordinate system. The rotation around Z-axis was omitted and set to 0). The Y-axis represents the rotation error (see Equation (4.19) for calculation) in degrees and ADD error (see Equation (4.17) for calculation), respectively. The 11 lines from bottom to top correspond to the 11 magnitudes we have for the error vectors, and they are encoded with the same colours in both plots. The specific colour-magnitude correspondence can be found in the legends of the plots.

From the figure, we also identified our most sensitive poses as rotation [-10,10,0] and rotation [50, -20, 0]. The exact value of the rotation error and the ADD error under these poses are listed in Table 5.1. The corresponding DRRs and mDRRs can be seen in Figure 5.5. Based on the identified poses, we continue with the magnifying-effect error analysis procedure.

Rotation	**[-10, 10, 0]**	**[50, -20, 0]**
Error magnitude	$10mm$ (image size $512mm \times 512mm$)	
Rotation error E_R (in degrees)	$25.3259°$	$6.5906°$
ADD error	$31.7525mm$	$45.2599mm$

Table 5.1: Identified sensitive poses and the corresponding error values.

Zoom-in on Sensitive Poses

After identifying the most sensitive poses, a similar error-introducing procedure for poses with rotation [50,-20,0] and [-10,10,0] was repeated, but from only adding an error to one of the landmarks to adding an error to all the landmarks. As can be seen in Figure 5.3 and 5.4, when the magnitude reaches $10mm$, the rotation error and the ADD error can be exceptionally bad. Based on the plots, when the error vector magnitudes are smaller than $3mm$, the mean ADD error can be kept under $10mm$. And the number of median rotation errors less than $5°$ generated by error vector magnitude less than $3mm$ is more than the number of under $5°$ rotation errors generated by error vector magnitude larger than $3mm$. For that reason, we decided to further investigate the magnitude of the error vector from $0mm$ to $3mm$ with an interval of $0.3mm$. We recorded the numerical values of the mean and median (50%) of the rotation error and the ADD error when the magnitude equals 3 (see Table 5.2 and Table 5.2). We also plotted the distribution of the changes in rotation error and ADD error under all magnitudes with box plots (see Figure 5.6, Figure 5.7, Figure 5.8 and Figure 5.9).

In Table 5.3, the highest ADD error was generated when 5 of the landmarks were introduced with error (landmarks with error introduced will be referred to as compromised points). As well as in Table 5.2, the median ADD error reached its highest when having 5 compromised landmarks. But attention needs to be paid towards the insignificant difference between the median ADD error with 5 compromised landmarks and the median ADD error with 6 compromised

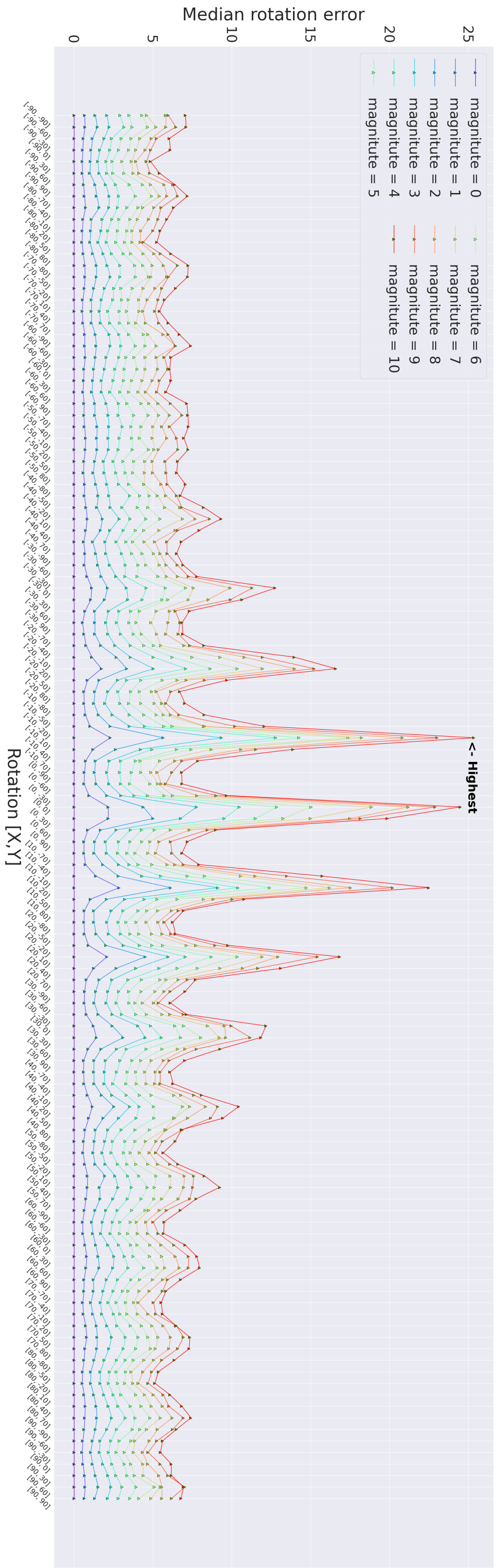


Figure 5.3: Mean rotation error

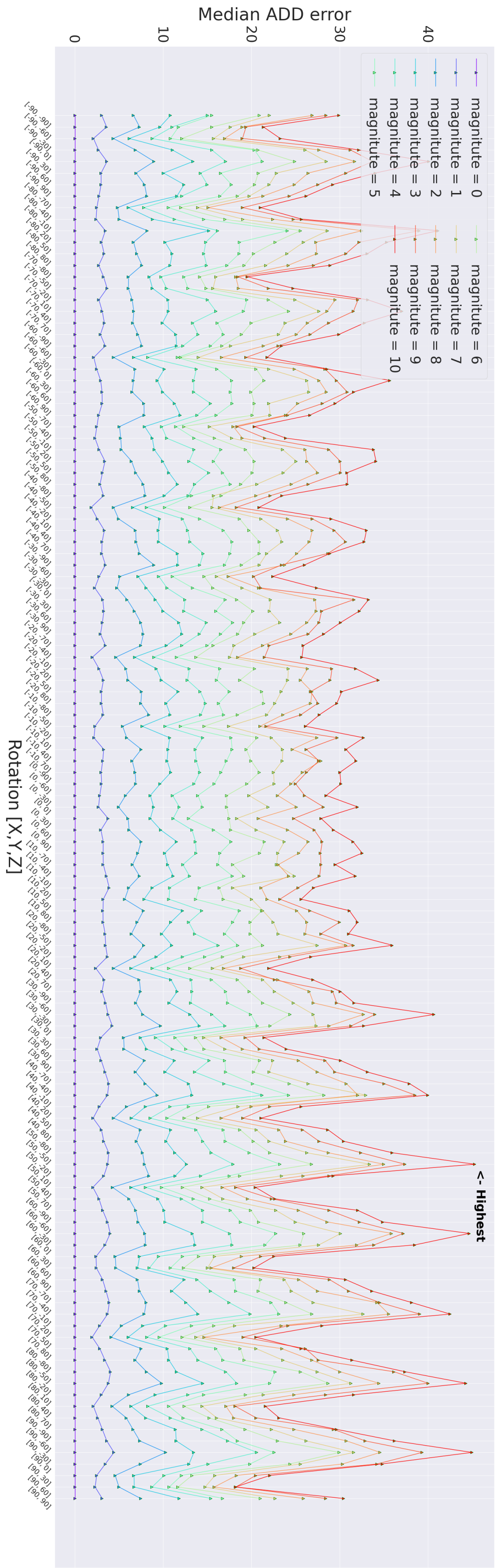
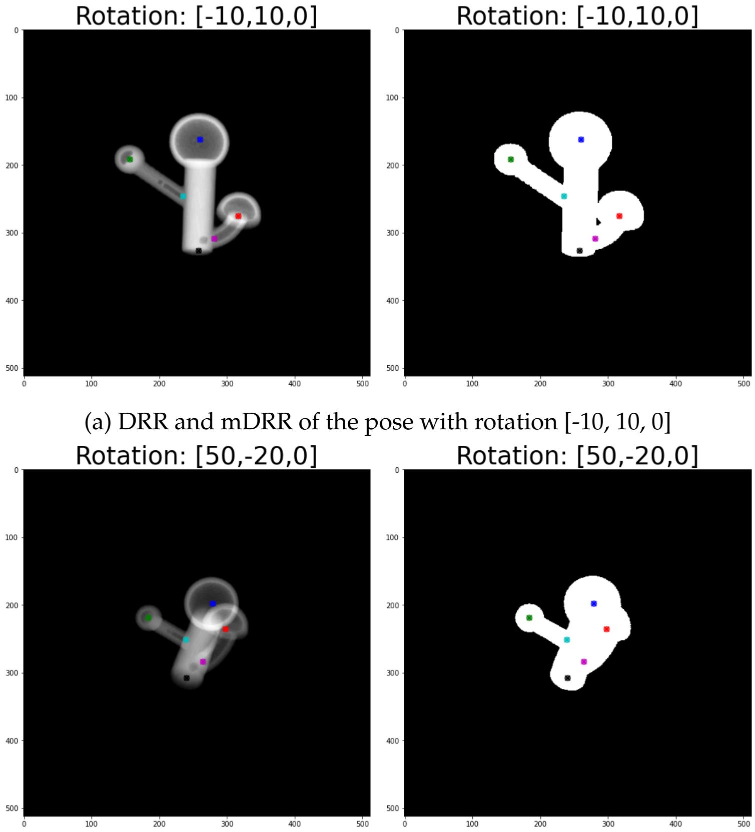


Figure 5.4: Mean ADD error



(a) DRR and mDRR of the pose with rotation [-10, 10, 0]

(b) DRR and mDRR of the pose with rotation [50, -20, 0]

Figure 5.5: DRRs and mDRRs of Sensitive Poses

landmarks. This could indicate that there's a potential for the median ADD error to reach the highest with 6 compromised landmarks and under rotation [50, -20, 0].

From Figure 5.6 and 5.8, the intention to keep at least 50% of the rotation error under $5°$ requires the magnitude for each error vector to be in $s_e \in [0, 1.8]$ ($6 \cdot s_e \in [0, 10.8]$ for 6 landmarks in total) on a 512×512 image, which approximately translates to $s_e \in [0, 1.125]$ ($6 \cdot s_e \in [0, 6.75]$ for 6 landmarks in total) on a 320×320 image and 0.35% of the image size. Yet, once we desire to have ADD error under 5, then from figure 5.7 and 5.9, a more strict requirement for the average magnitude of the error vector is formed, the average magnitude is forced down to $s_e \in [0, 1.5)(6 \cdot s_e \in [0, 9)$ for 6 landmarks in total) on a 512×512 image, which is $s_e \in [0, 0.9375)$ ($6 \cdot s_e \in [0, 5.625)$ for 6 landmarks in total) on a 320×320 image and takes up roughly 0.29296% of the image size. And this strict range is our identified model performance expectation:

$$\begin{gathered} P(E_{Euc} < 6 \cdot 0.293\% \cdot l_{DRR}) \geq 50\% \\ P(E_R < 5°) \geq 50\%, \\ P(ADD < 5mm) \geq 50\% \end{gathered} \tag{5.1}$$

where $\tilde{s_e}$ is the median of rotation error vector magnitude, and $l_{DRR} = w = h$ is the sidelength of the image. E_R is the rotation error, $E_{Euc} = \sum_{n=1}^{6} \|y_n, \tilde{y}_n\|_2$ is the sum of pairwise Euclidean distance between the ground truth and predicted 2D landmark coordinates.

Rotation		[50, -20, 0]					
# of compromised points		1	2	3	4	5	6
Magnitude		**3** (image size 512 × 512)					
Rotation error (in degrees)	Median	1.35550	1.81409	1.72348	1.89033	1.83287	2.03616
	Mean	1.35626	1.69417	1.84644	1.95795	1.90962	2.12439
ADD error	Median	10.43455	9.50278	8.41392	9.78926	11.64397	11.26135
	Mean	9.26265	11.13405	10.78351	12.21170	12.85408	13.45324

Table 5.2: The changes of the median, mean, first quartile and third quartile of rotation error and ADD error under pose [50, -20, 0] with error vector magnitude equals to 3 when introducing error to only one of the landmarks to introducing errors to all 6 of the landmarks.

Rotation		[-10, 10, 0]					
# of compromised points		1	2	3	4	5	6
Magnitude		**3** (image size 512 × 512)					
Rotation error (in degrees)	Median	5.09466	4.68021	7.08967	8.53233	8.15359	8.64018
	Mean	4.55804	7.49472	8.01562	10.97778	10.04391	10.63039
ADD error	Median	3.61111	5.09438	5.71610	7.47067	8.16992	6.77098
	Mean	3.67165	6.35236	7.40104	9.36920	9.65525	8.92902

Table 5.3: The changes of the median, mean, first quartile and third quartile of rotation error and ADD error under pose [-10, 10, 0] with error vector magnitude equals to 3 when introducing error to only one of the landmarks to introducing errors to all 6 of the landmarks

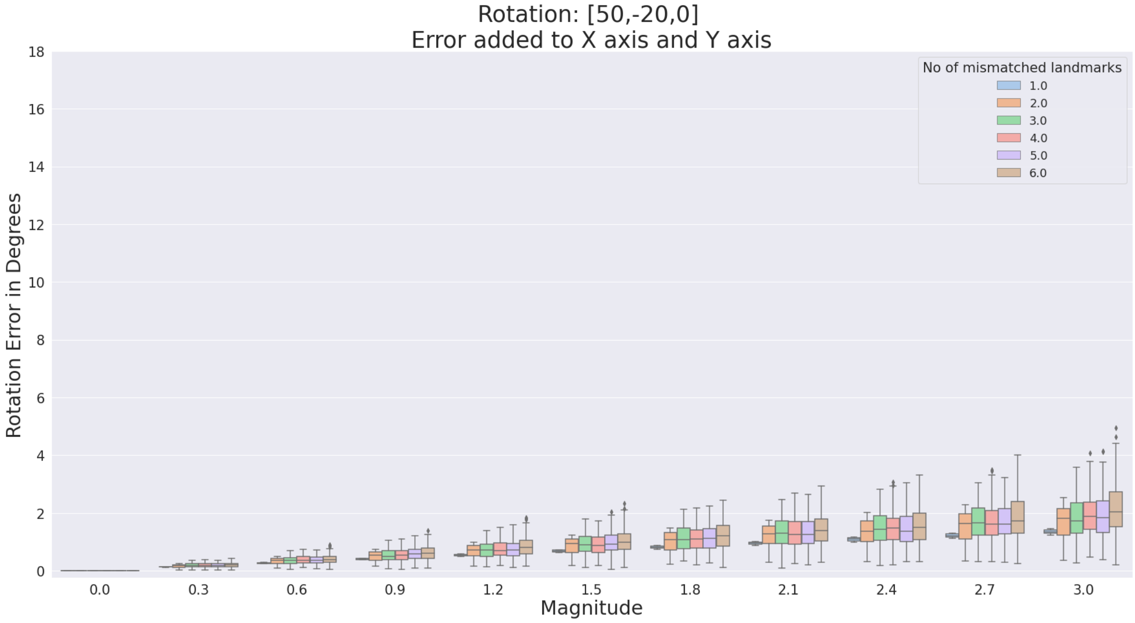


Figure 5.6: Rotation error under pose [50,-20], error magnitude ranges from 0 to 3 with 0.3 as interval

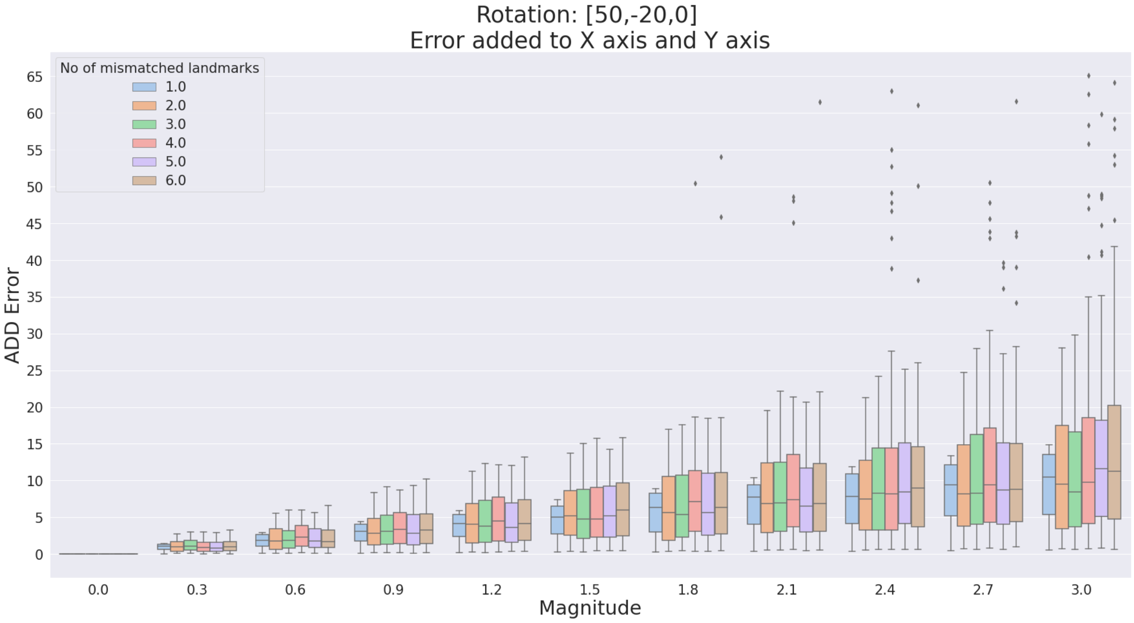


Figure 5.7: ADD error under pose [50,-20], error magnitude ranges from 0 to 3 with 0.3 as interval

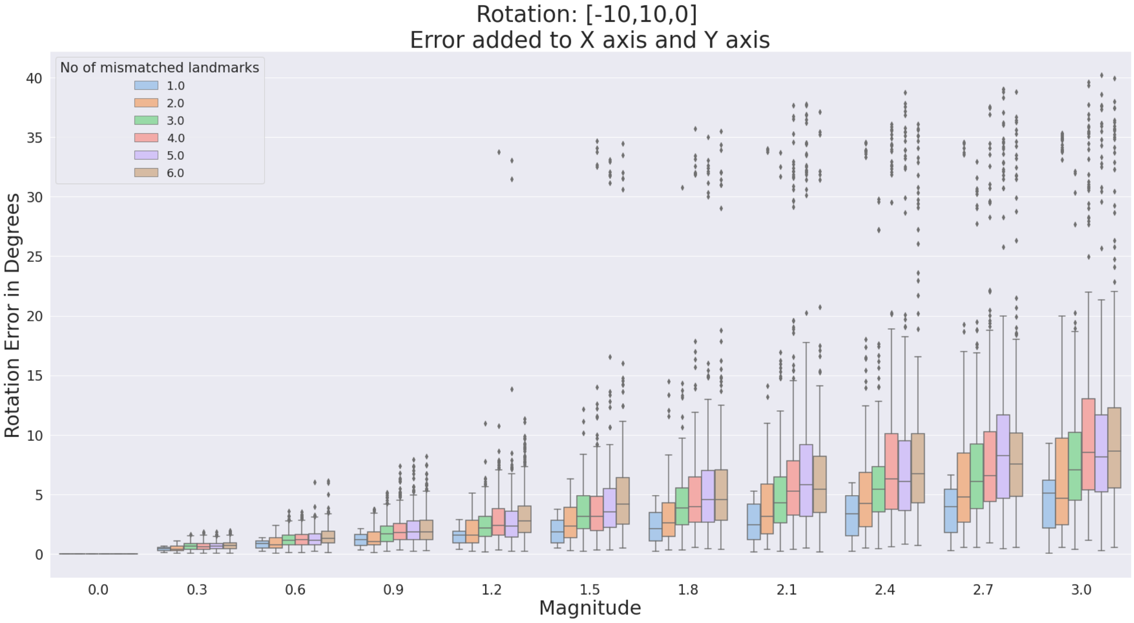


Figure 5.8: Rotation error under pose [-10,10], error magnitude ranges from 0 to 3 with 0.3 as interval

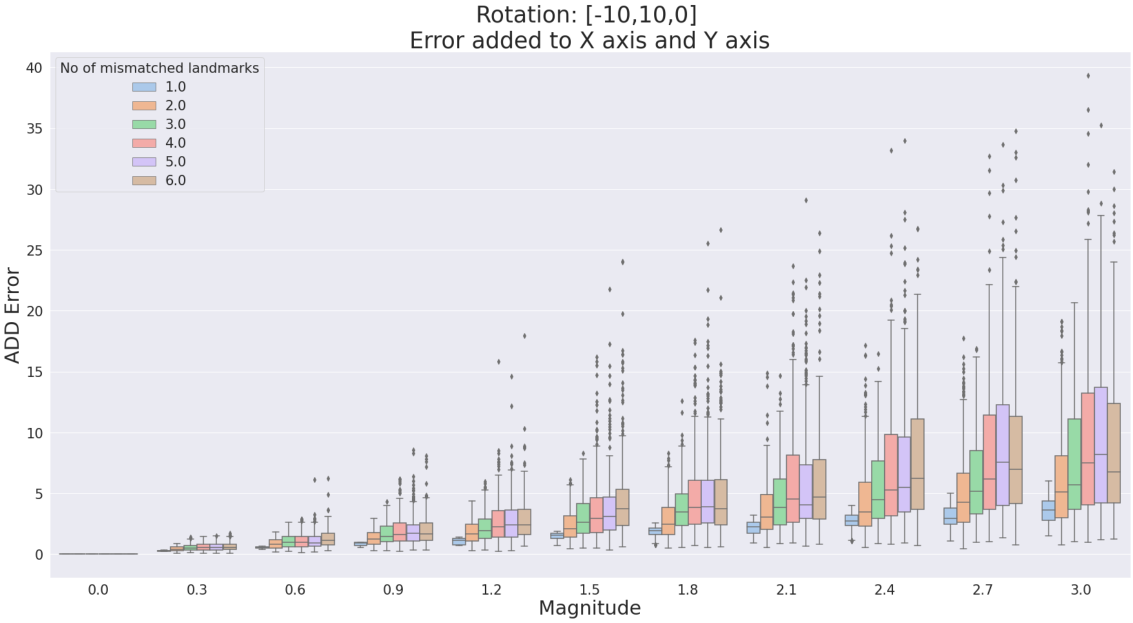


Figure 5.9: ADD error under pose [-10,10], error magnitude ranges from 0 to 3 with 0.3 as interval

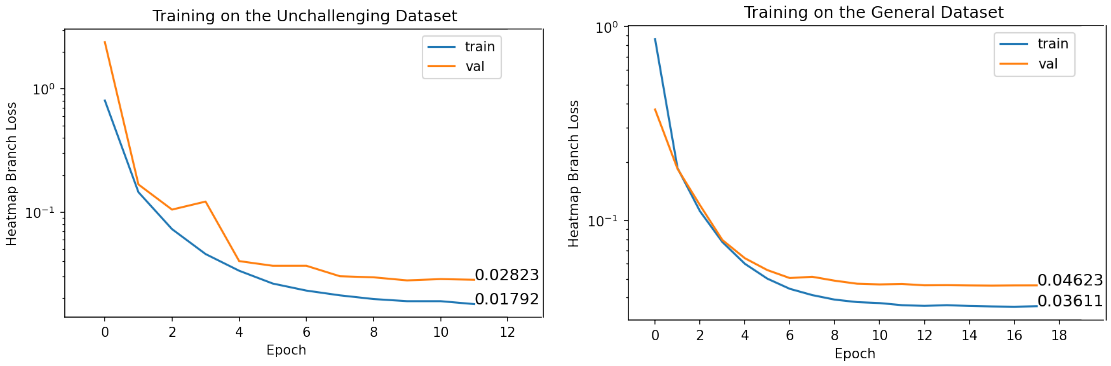
5.2.2 Model Training and Testing

Model Training

35 epochs were set for the training on both the general and unchallenging datasets with early stopping patience equal to 2. The network stopped after the 17th epoch when training with the general dataset, and it finished 11 epochs when training with the unchallenging dataset. The changes of loss of the localization branch (heatmap regression branch) on both the training set and validation set can be seen in Figure 5.10(a) and 5.10(b), overfitting of the training set was not prominent in both cases due to the use of early stopping. But the curves in both training processes flattened after the 7th epoch, which might be caused by the decay in the learning rate, which was decreased to $2.79936e{-}5$.

Model Testing

Afterwards, the model trained with the unchallenging dataset was tested on its very own corresponding test dataset that contains only unchallenging overlays, and the model trained with the general dataset was tested on both the general data test set and the test set dedicated for the training on the unchallenging data. Only the images with ground truth of all 6 landmarks lie within the frame that is $15mm$ from the image border (which results in a 290×290 frame region on the image), and all landmarks predicted were considered when calculating the Euclidean distance between the ground truth and the prediction. To illustrate the idea, the left image in Figure 5.11 represents the ones that were considered, and the image on the right side represents the ones that were discarded. The number of images with all landmarks within the 290×290 frame, as well as the number of 290×290 frames with all landmarks predicted, are listed in Table 5.4. Comparing the number of successfully predicted frames and the total frames considered, the model was capable of catching the attachment piece in most cases.



(a) Traing loss and validation loss when training on the unchallenging dataset

(b) Traing loss and validation loss when training on the challenging dataset

Figure 5.10: Training loss and validation loss on the unchallenging dataset and the general dataset

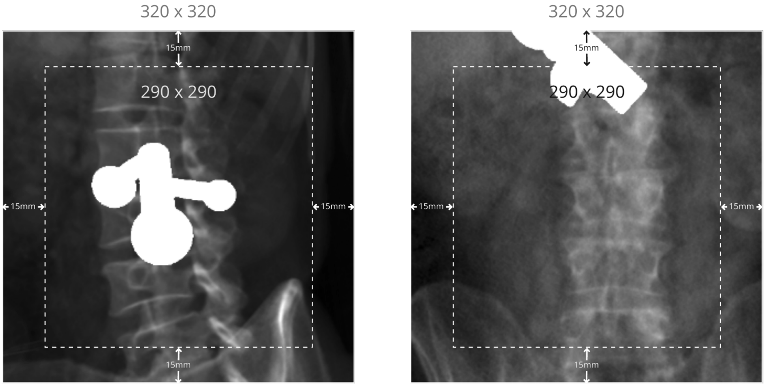


Figure 5.11: Examples of images that have been taken into consideration and images that have been ignored when calculating the Euclidean distance

Figure 5.12 shows the distribution of the sum of pairwise Euclidean distances between the ground truth and the predictions. Table 5.5 reports the test accuracy with regard to the pairwise Euclidean distance, we define the accuracy as the percentage of predicted images with 2D pairwise Euclidean distance E_{EUC} less than $6{\cdot}0.293\%{\cdot}l_{DRR}$. The median distance is the smallest when the model was trained on the unchallenging dataset and tested on the unchallenging dataset (corresponding to the last boxplot in Figure 5.12), and the value is $4.54439mm$, which means in 50% of the cases (median), on average per landmark has an approximately $0.75740mm$ deviation from its ground truth, and it takes up 0.2367% of the image size. As our assumption, the model train on the unchallenging dataset has the highest accuracy with a value of 69.37574% when testing on the unchallenging dataset. The median distance climbed up to a value of $4.84855mm$ when the model trained on the general dataset was tested on the unchallenging dataset, which means 50% of the case, on average, each landmark deviated from its ground truth 0.25253% of the image size. As can be seen in the last two box plots of Figure 5.12, the median Euclidean distance has a slight increase when the model traned on the general dataset tested on the unchallenging dataset, and a drop in accuracy can also be observed in Table 5.5.

It is certain that the general dataset contains a number of unchallenging images. To confirm

	General testset		Unchallenging testset	
# of images with all landmarks within the 290 x 290 frame	In total: 5258	Challenging: 3572	In total: 5109	
	Model trained on general dataset		Model trained on general dataset	Model trained on unchallenging dataset
# of 290 x 290 frames with all landmarks predicted	5251	3565	5104	5094

Table 5.4: Number of total considered images in each dataset and the corresponding number of successfully predicted images

our initial guess that the model did suffer from a performance decrease when dealing with challenging images, we further tested the model on challenging images from the general dataset. The results are shown as the second box plot in Figure 5.12. An increase in the median of the sum of pairwise Euclidean distance by around $0.40245mm$ comparing its performance on the general dataset can be seen. This could indicate that the challenging data is not as easy to be captured as the unchallenging data. Furthermore, the model stuffed from a significant performance decrease when tested on the challenging dataset compared to tested on the unchallenging dataset, the differences in median can reach up to $1.07174mm$. The median performance of both models on the unchallenging dataset fell into our initial requirement, but the ability of the model to handle challenging data still needs to be improved.

It is unfortunate that the ground truth rotation vector and translation vector were not recorded while the data generation step. However, from Figure 5.3 and 5.4, when the error vector magnitude equals 0 (i.e., no error was introduced to the 2D coordinates), the mean rotation error and the mean ADD error under the investigated rotations were approximately 0, which means the PnP algorithm is relatively stable when the landmarks processed no error. So we made a compromise. Instead of using the ground truth rotation matrix and translation vector obtained from QR decomposition, we used the PnP algorithm with the ground truth 2D landmark coordinates to calculate the near-ground truth rotation vector and translation vector. And we treated the near-ground truth rotation vector and translation vector as ground truth to calculate the rotation error and ADD error.

Figure 5.13 shows the distribution of the rotation error between the near-ground truth rotation vector and the rotation vector obtained from the predictions. Table 5.6 reports the test accuracy with regard to the rotation error, we define the accuracy as the percentage of predicted images with rotation satisfy rotation error E_R less than $5°$. From Figure 5.13, surprisingly, the median rotation error reached the highest when the model trained on the general dataset was tested on the unchallenging dataset, however, from Table 5.6, its percentage of predictions that has rotation error less than $5°$ is the second best. The rotation accuracies achieved by the model trained on the general dataset were similar. We also spotted distinct differences in rotation accuracies achieved by the two models. And the highest rotation accuracy was obtained by testing the model that trained on the unchallenging dataset with its own test set. Despite the accuracy gap, both models still exceeded our initial expectation, which is to have at least 50% of rotation error under $5°$.

Figure 5.14 shows the distribution of the ADD error between the reconstructed near-ground truth 3D landmarks using the near-ground truth rotation vector and translation vector and the reconstructed 3D landmarks using the rotation vector and translation vector obtained from the predictions. Table 5.7 reports the test accuracy with regard to the ADD error, we define the accuracy as the percentage of predicted images with reconstructed 3D landmarks that satisfy

ADD error less than $5mm$. The same as in rotation accuracy and Euclidean accuracy, the model trained on the unchallenging dataset had the best performance on its own test set, and the model trained on the general dataset had the poorest performance when testing on the challenging dataset. A more than 10% accuracy difference between the models' performance on the challenging dataset and unchallenging dataset could be another indication that under the current training setting and network structure, it is slightly difficult for the model to learn the features of the challenging images. Nevertheless, both models reached our expectation of the ADD accuracy, which is to have at least 50% of ADD error under $5mm$.

	Test accuracy (Euclidean distance)		
	On general dataset	On challenging images inside the general dataset	On unchallenging dataset
Model trained on the general dataset	51.45687%	45.32959%	63.96944%
Model trained on the unchallenging dataset	_	_	69.37574%

Table 5.5: Test accuracy regarding the pairwise Euclidean distance between prediction and ground truth

	Test accuracy (Rotation)		
	On general dataset	On challenging images inside the general dataset	On unchallenging dataset
Model trained on the general dataset	77.07103%	76.8864%	77.99765%
Model trained on the unchallenging dataset	_	_	81.6647%

Table 5.6: Test accuracy regarding the rotation error

	Test accuracy (ADD)		
	On general dataset	On challenging images inside the general dataset	On unchallenging dataset
Model trained on the general dataset	57.22719%	52.62272%	66.43809%
Model trained on the unchallenging dataset	_	_	71.86887%

Table 5.7: Test accuracy regarding the ADD error

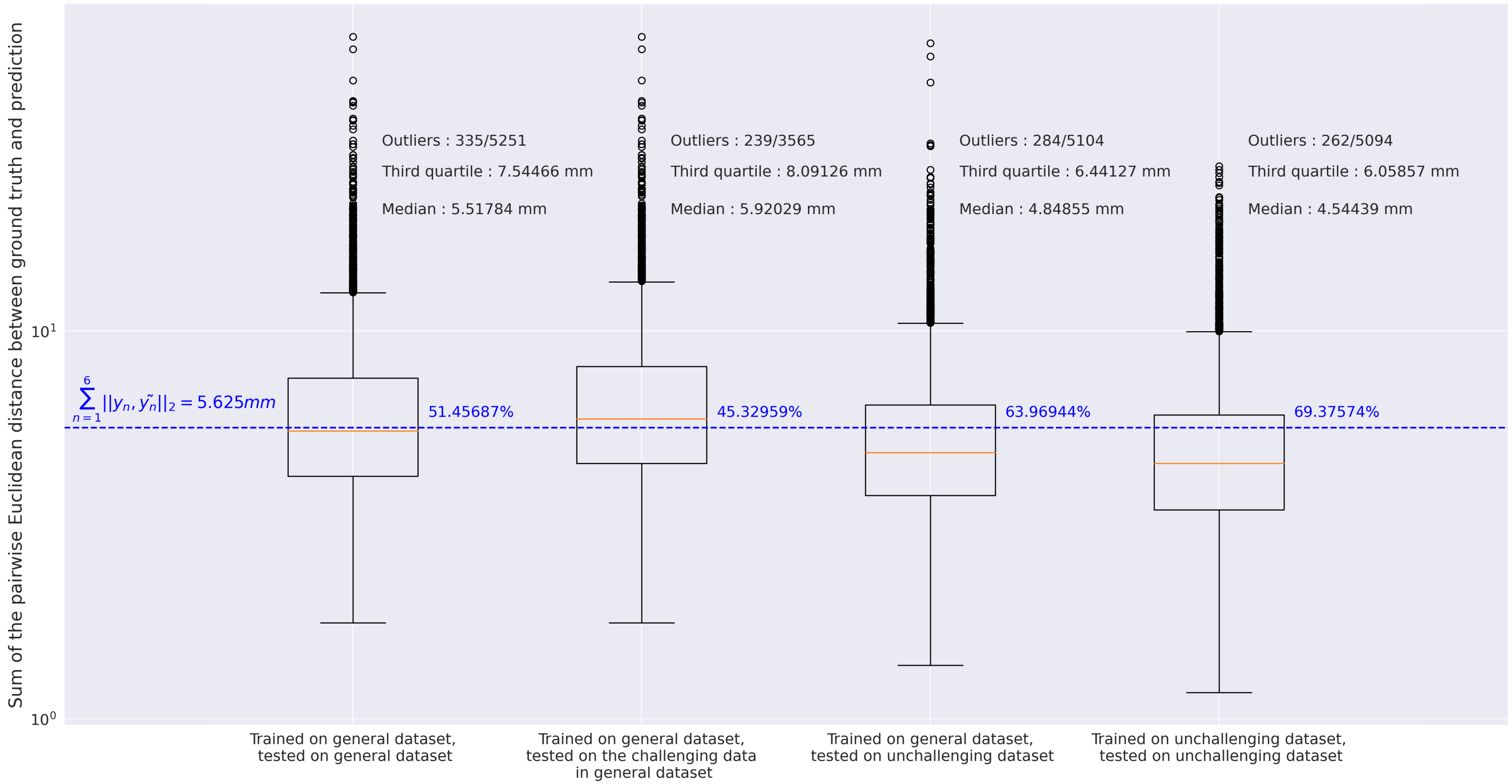


Figure 5.12: Distribution of the sum of pairwise Euclidean distances between the ground truth and the predictions when testing on different datasets

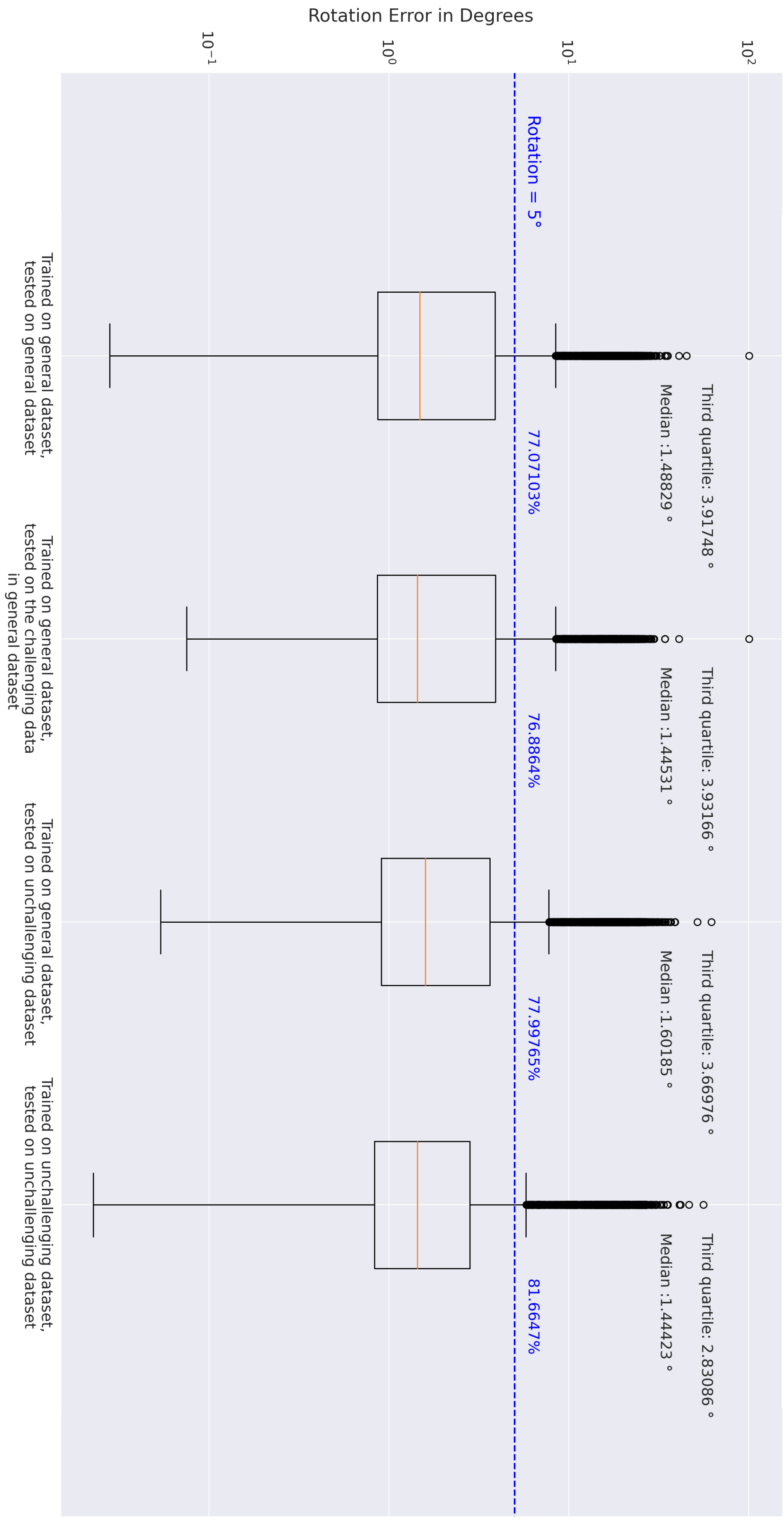


Figure 5.13: Distribution of the rotation error when testing on different datasets

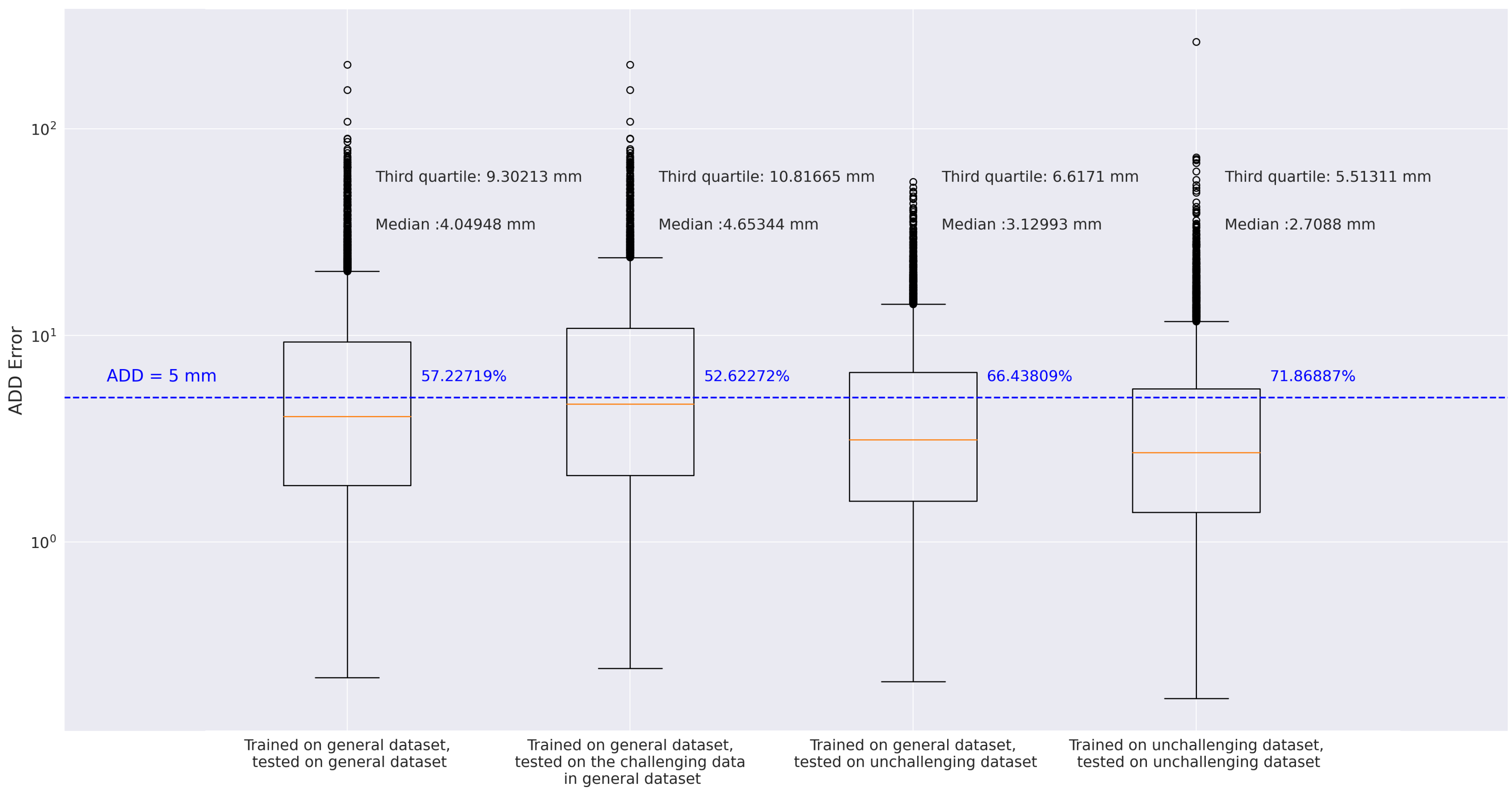


Figure 5.14: Distribution of the *ADD* error when testing on different datasets

Chapter 6

Discussion and Future Work

6.1 Discussion of the Results

Sensitive Poses

The near straight horizontal lines around 0 in both plots 5.3 and 5.4 when the magnitude of the errors equal to 0 in both figures may imply that the PNP algorithm performed stably while there is no error in the 2D coordinates. In Figure 5.3, the rotation errors start to increase dramatically when the poses of the attachment piece in 2D images approach its front view (when rotation equals [0, 0, 0]), and a distinct peak can be identified from this figure. In Figure 5.4, the lines are rather fluctuating, and with the magnitude getting larger and larger, the lines also become more and more volatile. Combining these two images, we also find that the rotation error reaching a peak doesn't necessarily invoke a peak in the ADD error, there are several cases when the rotation error reached the peak, but the ADD error was at the bottom.

Additionally, in Figure 5.4, there is also a relatively flat area in the centre of the figure, yet the rotation error is much more changeable, which could indicate that under some certain poses, the PnP algorithm was able to produce a translation vector which can compensate for the error introduced by the rotation, and the translation cannot be easily decoupled from rotation. A hypothesis is that the location of the landmarks on the 3D volume is not optimal, even though numerically, there is no collinear or coplanar landmark on our 3D volume, from the lateral view, the landmarks are indeed located within a small distance to a plane. However, the reason for this phenomenon is currently unclear. Nevertheless, if one values the ADD error over the rotation error, as the ADD error expresses the average distance between landmarks on the reconstructed object position and corresponding landmarks on the ground truth object position in the camera coordinate system, then using the generated unchallenging dataset to train an addition network as a filter to decide if the coming images contain the unchallenging poses and only perform landmark localization on the filtered-out unchallenging images could be an answer. However, this is certainly not an ideal solution to have in clinical practice.

Despite the fact that the PnP algorithm can be incredibly inaccurate when the magnitude reaches 10, its behaviour was relatively consistent as the most sensitive poses when error vector magnitude equals 1 are approximately the same as the ones where the magnitude equals 10 (i.e., the most sensitive poses remained the same regardless of the changes in the magnitude of the error vectors). And the good news is that the plots also show that the average error vector magnitude that we identified from the sensitive poses for the initial model performance exception range (ADD error under $5mm$ and rotation error under $5°$) is applicable for most of the rotations from [-90, -90, 0] to [90, 90, 0].

From table 5.2 and table 5.3, it is evident that with the increase in the number of compromised points, the PnP algorithm became more and more error-prone. But it is also worth noting that the median ADD error values did not always reach the highest when all landmarks had errors at the same time. An assumption is that when all the landmarks were introduced with errors, there was a chance that the added errors simulated a systematic error where all the error vectors were roughly pointing in the same direction, which caused a shift of the attachment piece in the 2D image, and when this happens, we would expect to have relatively small ADD error and rotation error compared to having errors pointing in heterogeneous directions.

Model Performance

Overall both trained models met our preliminary performance expectations. Although the model experienced an accuracy degradation when it was trained on the general dataset and tested on the challenging dataset, it failed to reach our expectation on the 2D pairwise Euclidean distance, yet the rotation error and ADD error it produced satisfied our expectations. When inspecting Figure 5.13 alone, two models performed relatively well and more than 70% predictions that produced less than $5\circ$ rotation error. But when combining the rotation accuracies reported in Table 5.6 with ADD accuracies reported in Table 5.7, considerably large accuracy differences (more than 10%) can be noticed. Based on Equation (4.20) for the calculation of ADD error, we inferred the contribution of the rotation error to the final ADD error might not be as significant as the error possessed by translation vectors. Thus it might implicitly confirm the observation we made when performing the error simulation that the effect of the rotation and the translation on the final pose cannot be easily decoupled. Yet if we compare the Euclidean distance accuracies reported in Table 5.5 with the ADD accuracies, the differences are not as prominent as the accuracy differences between the rotation accuracies and the ADD accuracies, the Euclidean distance accuracies were even lower than the ADD accuracies, which could suggest that the initial expectation we set for the model performance in the Euclidean distance was rather strict.

Additionally, we used 60000 images to train a model specifically for unchallenging poses, yet it only improved the Euclidean accuracy by around 5.4%, so the question arises: despite our observation that the PnP algorithm is more stable when facing errors under unchallenging poses, does it worth the effort to separate the poses for the neural network? It appears that the existence of challenging data in the training set did not jeopardize the model's ability to learn the front-view-specific features to a hazardous extent. However, the model's confusion on the challenging images complies with our initial assumption. It is evident that the separation of challenging poses and unchallenging poses benefited the model's performance on unchallenging images. But if we increase the proportion of unchallenging images in the general dataset and feed the model with more data, will the performance of the model on the unchallenging dataset match that of the model trained exclusively for unchallenging images? The hypothesis is positive. However, as the name suggested, we would like to have the model trained on the general dataset to be able to handle both kinds of images well. Therefore, finding an appropriate balance between the number of unchallenging images and the number of challenging images in the general dataset so that the model achieves satisfying accuracy on both kinds of images needs to be placed under investigation.

Furthermore, when we started the training of the models, the initial learning rate was set to 0.001, which might be a comparably large starting learning rate even with the decay we applied. The nearly invisible loss decrease in the last epochs in both training processes might have been caused by a plateau area, a saddle point, a local minimum, or in the best case, a global optimum, but it is unclear which scenario it fell into. Thus we cannot yet give a verdict either on whether the decay coefficient was appropriate or on whether the decayed learning rate was too small or too large. It requires further effort to explore the suitable initial learning rates and decay methods.

From what we have observed until now, the models can at least meet our initial expectations

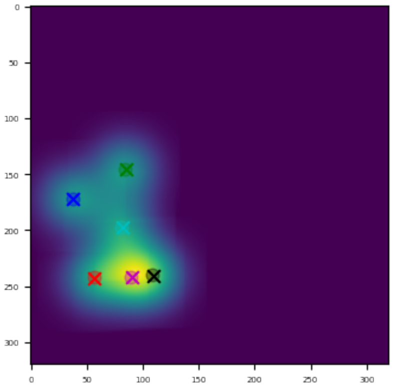


Figure 6.1: Example of post-process inaccuracy

Ground truth coordinates		After post-processing		Euclidean distance	Distance sum
X	Y	X	Y		
37.1829	171.5434	37.7388	172.0551	0.7556	4.3858
84.1942	145.2641	84.7348	145.8088	0.7674	
56.5005	242.3362	57.	242.8504	0.7169	
81.3708	197.1330	81.8959	197.6716	0.7522	
89.5942	241.3432	90.0848	241.8716	0.7211	
109.0639	240.2676	108.5770	239.8035	0.6727	

Table 6.1: Corresponding ground truth and after post-processing coordinates, and distances between them (after rounding).

when dealing with unchallenging poses, but in our pipeline, there exists a weak point, which is the post-processing process. Even though we fed the post-processing with heatmaps generated based on the ground truth coordinates, the resulting centre coordinates of the blobs were not exactly equal to the ground truth coordinates, the differences between them are visually small on a 320×320 images, but the sum of the pairwise Euclidean distances can be prominent. An example can be seen in Figure 6.1, in which the background is the sum of 6 ground truth heatmaps with a standard deviation equal to 10, and the ground truth coordinates are marked with circles while the after-post-processing coordinates are marked with crosses, the numerical values of the coordinates and distances between them can be seen in Table 6.1. Our assumption is there are two possible elements involved in this pipeline that may contain small errors: the first one is that the cropping operation we performed when setting the background area to 0 affected the blob detection's ability to detect circles, and when performing the blob detection, the detected blobs cannot perfectly cover the heatmap regions which leads to slightly skewed centres; another one is that the parameters set for the blob detection that control the shape of the blob were not optimal.

6.2 Possible Future Work

A further investigation into the reason for the certain behaviour of the PnP algorithm we observed is needed, which will be a valuable input to the design of the next version attachment piece. The criteria we set for separating unchallenging cases from challenging cases are not sophisticated enough. It only works when the translation along Z-axis is zero, it takes an extra image generation time and memory to make the decision, which this not ideal. However, we can still use the widgets and criteria to produce enough unchallenging data and then use the data to train a classification network.

The influence of the use of other loss functions is also worth examining, especially the AdaLoss (Teixeira et al., 2019), which was specifically introduced for landmark localization through heatmap regression. As mentioned above, the model might need a better initial learning rate and decay coefficient to take its performance to the next level. Thus, the impact of different initial learning rates, different learning rate schedule methods, as well as different optimizers on the model's convergence is also another direction to look into. The change in the backbone model can also be considered. Moreover, as one may have noticed, our model's structure is relatively simple and straightforward. Even though it has proven our idea that using intraoperative fluoroscopy to guide the intervention is a feasible approach, however, to able become a real-world clinical application, the requirement for accuracy is much higher. Studies that have been done by Gundle et al. (2017) on the accuracy and precision of a current surgical navigation system reported a lesser than $0.25mm$ RMS (Root Mean Square) error between the navigation system-calculated distance and the ground truth distance when simulating tracking on a machined grid. Therefore, in order to be able to compete with the current optical surgical navigation systems in the future, further investigation needs to be done on more advanced neural network structures in order to reach a higher accuracy than the current accuracy provided by the proposed model.

Image resolution is another factor that may have an impact on the model's performance, in this project, the model was trained with images in size 320×320, which is a rather small size. The real fluoroscopy image sizes can range from 512×512 to 2048×2048[1]. Sabottke and Spieler (2020) observed a performance gain when training CNNs (Convolutional Neural Networks) with an increased resolution of radiographic images. Hence, the effect of image resolution on the proposed model can be worthwhile to inspect.

A multiplication operation may be more informative than the current addition one in the sense that by multiplication, the segmentation can black out the background area for the localization branch, which intuitively has a chance to lead to better performance. Possible ways how to utilize the information from the segmentation branch more efficiently are also worth exploring. Besides, the current model was trained with only the attachment piece and anatomy within the field of view, which means we are not insured when additional surgical instruments enter the image frame, and to be able to deploy the pipeline in the real world, the further investigation on the impact of having other structures in the images is necessary.

There is another usage of such landmark detection network that is interesting to explore, namely, to serve as a guide for medical image registration. Medical image registration is a process of aligning different image modalities together. There are many methods that can perform automatic registration, but the inherent issue is that if two images are far away from each other, the algorithms are prone to stick at the local optimal, yet with landmark detection, we can use attachment piece to explicitly mark out the correspondence between the image modalities and bring them together to a place that near global optimal and then the automatic registration can take over from there.

[1] https://siim.org/page/archiving_chapter2

Chapter 7

Conclusion

This project tried to find a solution to help facilitate the realization of the newly emerged surgical navigation idea X23D.

The possibility of utilizing neural networks in aiding the surgical instruments' pose estimation task from a single DRR has been investigated. Ideas have been borrowed from the existing surgical navigation systems, instead of locating instruments directly, a reference frame (attachment piece) which serves as a medium has been introduced to implicitly indicate the location and the pose of the surgical instruments. The attempt to first find the landmarks of the reference frame in DRRs and then compute its pose using the PnP algorithm in the 3D world has been made. In order to establish an impression of how 2D landmarks' coordinates error impact its final calculated 3D pose, an error simulation procedure has been carried out. From the simulation results, an initial performance expectation has been formed exclusively for this project, and for other methods that will be examined to locate the attachment piece in 2D images for X23D in the future. Criteria which are able to partially separate the 2D near-front-view poses of the attachment piece from other poses have been proposed. Datasets for the testing and validation of the neural network have been generated with the help of the criteria. A neural network structure that is able to fulfil the initial performance expectation has been designed and trained. The results proved that it is auspiciously possible to locate the attachment piece in the 3D world by using only one DRR with only 6 predefined landmarks. But to be able to apply the proposed method to real-world applications, the accuracy still needs to be improved, and the robustness when having surgical instruments appear in the image needs to be analysed. Yet this work set a stepping stone for deeper explorations. It can be seen as a promising starting point for further researchers to employ neural networks to provide surgical navigation.

List of Figures

List of Tables

Bibliography

Acuna, R. and Willert, V. (2018). Insights into the robustness of control point configurations for homography and planar pose estimation. *arXiv preprint arXiv:1803.03025*.

Bier, B., Unberath, M., Zaech, J.-N., Fotouhi, J., Armand, M., Osgood, G., Navab, N., and Maier, A. (2018). X-ray-transform invariant anatomical landmark detection for pelvic trauma surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 55–63. Springer.

Boudissa, M., Carmagnac, D., Kerschbaumer, G., Ruatti, S., and Tonetti, J. (2022). Screw misplacement in percutaneous posterior pelvic iliosacral screwing with and without navigation: A prospective clinical study of 174 screws in 127 patients. *Orthopaedics & Traumatology: Surgery & Research*, 108(2):103213.

Bui, M., Albarqouni, S., Schrapp, M., Navab, N., and Ilic, S. (2017). X-ray PoseNet: 6 DoF pose estimation for mobile X-ray devices. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1036–1044. IEEE.

Ding, L., Zheng, K., Lin, D., Chen, Y., Liu, B., Li, J., and Bruzzone, L. (2021). MP-ResNet: Multipath residual network for the semantic segmentation of high-resolution polsar images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

Do, T.-T., Cai, M., Pham, T., and Reid, I. (2018). Deep-6dpose: Recovering 6d object pose from a single RGB image. *arXiv preprint arXiv:1802.10367*.

Dorgham, O., Laycock, S., and Fisher, M. (2012). GPU accelerated generation of digitally reconstructed radiographs for 2D/3D image registration. *IEEE Transactions on Biomedical Engineering*, 59(9):2594–2603.

Dosovitskiy, A., Tobias Springenberg, J., and Brox, T. (2015). Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1538–1546.

Duffhauss, F., Demmler, T., and Neumann, G. (2022). MV6D: Multi-view 6D pose estimation on RGB-D frames using a deep point-wise voting network. *arXiv preprint arXiv:2208.01172*.

Fard, A. P., Ferrantelli, J., Dupuis, A.-L., and Mahoor, M. H. (2022). Sagittal cervical spine landmark point detection in X-ray using deep convolutional neural networks. *IEEE Access*.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Foley, K. T., Simon, D. A., and Rampersaud, Y. R. (2000). Virtual fluoroscopy. *Operative Techniques in Orthopaedics*, 10(1):77–81.

Gamra, M. B. and Akhloufi, M. A. (2021). A review of deep learning techniques for 2D and 3D human pose estimation. *Image and Vision Computing*, 114:104282.

Gao, C., Unberath, M., Taylor, R., and Armand, M. (2019). Localizing dexterous surgical tools in x-ray for image-based navigation. *arXiv preprint arXiv:1901.06672*.

Goldberg, J. L., Kirnaz, S., Carnevale, J. A., McGrath, L., and Härtl, R. (2022). History of navigation guided spine surgery. In *Technical Advances in Minimally Invasive Spine Surgery*, pages 3–10. Springer.

Gundle, K. R., White, J. K., Conrad, E. U., and Ching, R. P. (2017). Suppl-3, m4: Accuracy and precision of a surgical navigation system: Effect of camera and patient tracker position and number of active markers. *The open orthopaedics journal*, 11:493.

Guo, F., He, Y., and Guan, L. (2017). RGB-D camera pose estimation using deep neural network. In *2017 IEEE global conference on signal and information processing (GlobalSIP)*, pages 408–412. IEEE.

Halm, J. A., Beerekamp, M. S. H., de Muinck-Keijzer, R. J., Beenen, L. F., Maas, M., Goslings, J. C., and Schepers, T. (2020). Intraoperative effect of 2D vs 3D fluoroscopy on quality of reduction and patient-related outcome in calcaneal fracture surgery. *Foot & ankle international*, 41(8):954–963.

Hatabu, H. and Madore, B. (2021). Dark-field chest x-ray imaging: an evolving technique in the century-old history of chest x-ray imaging. *Radiology*, 301(2):396–397.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K. and Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

He, Z., Feng, W., Zhao, X., and Lv, Y. (2020). 6D pose estimation of objects: Recent technologies and challenges. *Applied Sciences*, 11(1):228.

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N. (2012). Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer.

Hu, Y., Fua, P., Wang, W., and Salzmann, M. (2020). Single-stage 6D object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939.

Hu, Y., Hugonot, J., Fua, P., and Salzmann, M. (2019). Segmentation-driven 6D object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3385–3394.

Hussain, I., Cosar, M., Kirnaz, S., Schmidt, F. A., Wipplinger, C., Wong, T., and Härtl, R. (2020). Evolving navigation, robotics, and augmented reality in minimally invasive spine surgery. *Global Spine Journal*, 10(2_suppl):22S–33S.

Hussain, I., Navarro-Ramirez, R., Lang, G., and Härtl, R. (2018). 3D navigation-guided resection of giant ventral cervical intradural schwannoma with 360-degree stabilization. *Clinical spine surgery*, 31(5):E257–E265.

Jadon, S. (2020). A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE.

Janssen, I., Lang, G., Navarro-Ramirez, R., Jada, A., Berlin, C., Hilis, A., Zubkov, M., Gandevia, L., and Härtl, R. (2017). Can fan-beam interactive computed tomography accurately predict indirect decompression in minimally invasive spine surgery fusion procedures? *World Neurosurgery*, 107:322–333.

Kang, J., Oh, K., and Oh, I.-S. (2021). Accurate landmark localization for medical images using perturbations. *Applied Sciences*, 11(21):10277.

Kendall, A., Grimes, M., and Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946.

Kułakowski, M., Reichert, P., Elster, K., Witkowski, J., Ślęczka, P., Morasiewicz, P., Oleksy, Ł., and Królikowska, A. (2022). Differences in accuracy and radiation dose in placement of iliosacral screws: Comparison between 3D and 2D fluoroscopy. *Journal of Clinical Medicine*, 11(6):1466.

Laina, I., Rieke, N., Rupprecht, C., Vizcaíno, J. P., Eslami, A., Tombari, F., and Navab, N. (2017). Concurrent segmentation and localization for tracking of surgical instruments. In *International conference on medical image computing and computer-assisted intervention*, pages 664–672. Springer.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. *International journal of computer vision*, 81(2):155–166.

Li, Z., Wang, G., and Ji, X. (2019). CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Lu, X. X. (2018). A review of solutions for perspective-n-point problem in camera pose estimation. In *Journal of Physics: Conference Series*, volume 1087, page 052009. IOP Publishing.

Lu, Y., Qin, X., Fan, H., Lai, T., and Li, Z. (2021). Wbc-net: A white blood cell segmentation network based on unet++ and resnet. *Applied Soft Computing*, 101:107006.

Madeja, R., Pometlová, J., Osemlak, P., Voves, J., Bialy, L., Vrtková, A., and Pleva, L. (2022). Comparison of fluoroscopy and fluoroscopy-based 2D computer navigation for iliosacral screw placement: a retrospective study. *European Journal of Trauma and Emergency Surgery*, pages 1–6.

Malham, G. M. and Wells-Quinn, T. (2019). What should my hospital buy next?—guidelines for the acquisition and application of imaging, navigation, and robotics for spine surgery. *Journal of Spine Surgery*, 5(1):155.

Mendelsohn, D., Strelzow, J., Dea, N., Ford, N. L., Batke, J., Pennington, A., Yang, K., Ailon, T., Boyd, M., Dvorak, M., et al. (2016). Patient and surgeon radiation exposure during spinal instrumentation using intraoperative computed tomography-based navigation. *The Spine Journal*, 16(3):343–354.

Merloz, P., Troccaz, J., Vouaillat, H., Vasile, C., Tonetti, J., Eid, A., and Plaweski, S. (2007). Fluoroscopy-based navigation system in spine surgery. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 221(7):813–820.

Mezger, U., Jendrewski, C., and Bartels, M. (2013). Navigation in surgery. *Langenbeck's archives of surgery*, 398(4):501–514.

Montúfar, J., Romero, M., Muñoz-Jiménez, V., Scougall-Vilchis, R., and Jiménez, B. (2018). Perspective and orthogonal CBCT/CT digitally reconstructed radiographs compared to conventional cephalograms. In *Conf. Biomedical Engineering and Sciences | BIOENG'16*, pages 41–45.

Navarro-Ramirez, R., Lang, G., Lian, X., Berlin, C., Janssen, I., Jada, A., Alimi, M., and Härtl, R. (2017). Total navigation in spine surgery; a concise guide to eliminate fluoroscopy using a portable intraoperative computed tomography 3-dimensional navigation system. *World neurosurgery*, 100:325–335.

Noothout, J. M., de Vos, B. D., Wolterink, J. M., Leiner, T., and Išgum, I. (2018). Cnn-based landmark detection in cardiac cta scans. *arXiv preprint arXiv:1804.04963*.

Noothout, J. M., De Vos, B. D., Wolterink, J. M., Postma, E. M., Smeets, P. A., Takx, R. A., Leiner, T., Viergever, M. A., and Išgum, I. (2020). Deep learning-based regression and classification for automatic landmark localization in medical images. *IEEE transactions on medical imaging*, 39(12):4011–4022.

Oberweger, M., Rad, M., and Lepetit, V. (2018). Making deep heatmaps robust to partial occlusions for 3D object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. (2018). Attention U-Net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Otomo, N., Funao, H., Yamanouchi, K., Isogai, N., and Ishii, K. (2022). Computed tomography-based navigation system in current spine surgery: A narrative review. *Medicina*, 58(2):241.

Pachón, C. G., Ballesteros, D. M., and Renza, D. (2021). Fake banknote recognition using deep learning. *Applied Sciences*, 11(3):1281.

Payer, C., Štern, D., Bischof, H., and Urschler, M. (2016). Regressing heatmaps for multiple landmark localization using CNNs. In *International conference on medical image computing and computer-assisted intervention*, pages 230–238. Springer.

Presenti, A., Liang, Z., Pereira, L. F. A., Sijbers, J., and De Beenhouwer, J. (2022). Fast and accurate pose estimation of additive manufactured objects from few x-ray projections. *Expert Systems with Applications*, page 118866.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Rad, M. and Lepetit, V. (2017). Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3836.

Riegler, G., Urschler, M., Ruther, M., Bischof, H., and Stern, D. (2015). Anatomical landmark detection in medical applications driven by synthetic data. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 12–16.

Ritschl, L., Kuntz, J., and Kachelrieß, M. (2015). The rotate-plus-shift c-arm trajectory: complete ct data with limited angular rotation. In *Medical Imaging 2015: Physics of Medical Imaging*, volume 9412, pages 497–500. SPIE.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Sabottke, C. F. and Spieler, B. M. (2020). The effect of image resolution on deep learning in radiography. *Radiology. Artificial intelligence*, 2(1).

Silberman, N., Sontag, D., and Fergus, R. (2014). Instance segmentation of indoor scenes using a coverage loss. In *European conference on computer vision*, pages 616–631. Springer.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer.

Świątek-Najwer, E., Będziński, R., Krowicki, P., Krysztoforski, K., Keppler, P., and Kozak, J. (2008). Improving surgical precision–application of navigation system in orthopedic surgery. *Acta of Bioengineering and Biomechanics*, 10(4):55–62.

Teixeira, B., Tamersoy, B., Singh, V., and Kapoor, A. (2019). Adaloss: Adaptive loss function for landmark localization. *arXiv preprint arXiv:1908.01070*.

Torres, J., James, A. R., Alimi, M., Tsiouris, A. J., Geannette, C., and Härtl, R. (2012). Screw placement accuracy for minimally invasive transforaminal lumbar interbody fusion surgery: a study on 3-d neuronavigation-guided surgery. *Global spine journal*, 2(3):143–151.

Toshev, A. and Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660.

Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., and Savarese, S. (2019). Densefusion: 6D object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732.

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2017). Posecnn: A convolutional neural network for 6D object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*.

Zhang, J., Liu, M., and Shen, D. (2017). Detecting anatomical landmarks from limited medical imaging data using two-stage task-oriented deep neural networks. *IEEE Transactions on Image Processing*, 26(10):4753–4764.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890.

Zhong, Z., Li, J., Zhang, Z., Jiao, Z., and Gao, X. (2019). An attention-guided deep regression model for landmark detection in cephalograms. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 540–548. Springer.