Automatic Landmark Identification in 3D Cone-Beam Computed Tomography scans

Maxime Gillot[1] MAX.GILLOT.69@GMAIL.COM
Baptiste Baquero[1] BBAQUERO@UMICH.EDU
Antonio Ruellas[1,2] ARUELLAS@UMICH.EDU
Marcela Gurgel[1] MLIMAGUR@UMICH.EDU
Elizabeth Biggs[1] EELNER@UMICH.EDU
Marilia Yatabe[1] MSYATABE@UMICH.EDU
Jonas Bianchi[1,3] BIANCHIJ@UMICH.EDU
Lucia Cevidanes[1] LUCIACEV@UMICH.EDU
Juan Carlos Prieto[4] JPRIETO@MED.UNC.EDU

[1] *University of Michigan, Ann Arbor, USA*
[2] *Federal University of Rio de Janeiro, Rio de Janeiro, Brazil*
[3] *University of the Pacific, San Francisco, CA, USA*
[4] *University of North Carolina, Chapel Hill, NC, USA*

Editors: Under Review for MIDL 2022

Abstract

Robust and accurate solutions for anatomical landmark detection support entire clinical workflows from diagnosis, therapy planning, intervention and follow-up, image-to-image registration, structure tracking and simulations. In this paper, we propose a novel approach that reformulates landmark detection as a classification problem through a virtual agent placed inside a 3D Cone-Beam Computed Tomography (CBCT) scan. This agent is trained to navigate in a multi-scale volumetric space to reach the estimated landmark position. The agent movements decision relies on a combination of Densely Connected Convolutional Networks (DCCN) and fully connected layers. We evaluated our approach on 60 CBCT scans from teenagers to senior patients. For each CBCT, 34 ground truth landmark positions were identified by clinicians. Our method achieved a high accuracy with an average of 1.21±0.79 mm error on the 6 landmark positions without failures. Moreover, it takes an average of 25.2s computation time to identify 6 landmarks on one large 3D-CBCT scans using GPU .

Keywords: Deep learning, Agent based learning, Medical image analysis, Multi-scale images, Three-dimensional landmark identification, Smart localization.

1. Introduction

The accurate anatomical landmarks localization for medical imaging data is a challenging problem due to frequent ambiguity of their appearance and the rich variability of the anatomical structures. Landmark detection represents a prerequisite for medical image

analysis. It supports entire clinical workflows from diagnosis (Zhang et al., 2017), therapy planning (Yu et al., 2013), intervention, follow-up to structure tracking (Yang et al., 2011) and simulations (Cebral and Lohner, 2005). Landmark identification may serve as initialization to other algorithms such as instance segmentation algorithms (Pouch et al., 2014), or image-to-image registration (Lüthi et al., 2011),(Glocker et al., 2014). Figure 1 shows different landmarks placed on CBCT scans. Most of the available solutions for landmark detection rely on machine learning (Ghesu et al., 2016), (Donner et al., 2013), (Cuingnet et al., 2012). Other approaches for landmark identification rely on sub-optimal search strategies, *i.e.*, exhaustive scanning (Ghesu et al., 2016; Donner et al., 2013), one-shot displacement estimation (Criminisi et al., 2010), (Štern et al., 2016), or end-to-end image mapping techniques (Dai et al., 2016; Payer et al., 2016). In many cases these methods can lead to false-positive detection results and excessively high computation times. In this work, we present a new method inspired by a deep reinforcement learning technique (Ghesu et al., 2017). Our approach is robust and finds landmarks in CBCT scans accurately and automatically, hence assisting clinicians in this crucial but time-consuming task. The landmark detection task is setup as a behavior classification problem for an artificial agent which is navigating through the voxel-grid of the image at different spatial resolutions. The detection starts at a low-resolution image with a global context, and continues at the higher-resolution image capturing increased levels of detail. The image features are used as indicators to guide the landmark search. Figure 2 shows a CBCT image at different resolution levels, these are the agent's environment. In order to adapt the feature extraction, we train different neural networks at each resolution. After the feature extraction our search model takes as input the image features and decides in which direction the agent should move as shown in Figure 3. The search model is a classification model that uses a features extraction network and fully connected layers. In the following section we describe the images used to train our agent, followed by related work on approaches to find landmarks in medical images.

2. Materials

The Cone-Beam Computed Tomography scans (CBCT) (Shah et al., 2014) were acquired on teenagers to senior patients for dental clinical purposes. The CBCTs contain a diversity of bones structures in a total of 60 scans. Forty of the CBCTs were performed at a private dental radiology clinic using the i-CAT device (Imaging Sciences International, Hatfield, PA), configured with 120 Kvp, 3–8 mA, a 0.4-mm isometric voxel size, and a field of view (FOV) of 23 cm $\times$ 17 cm. The other 20 CBCTs were acquired in another radiology clinic with the same device but a 0.3-mm isometric voxel size, and a field of view (FOV) of 17 cm $\times$ 17 cm. Two open-source software packages, ITK-SNAP 3.8 (Yushkevich et al., 2006) and Slicer 4.11 (Fedorov et al., 2012) were used to orient the scans and place the 34 landmarks. In this paper, we selected a set of 6 different landmarks located in different types of anatomical structures, including, bone, tooth and non-rigid organ (Figure 1). This set was selected to test the robustness of the solutions when exposed to a variation of contrast, shape and position of the anatomical structures in the scan.

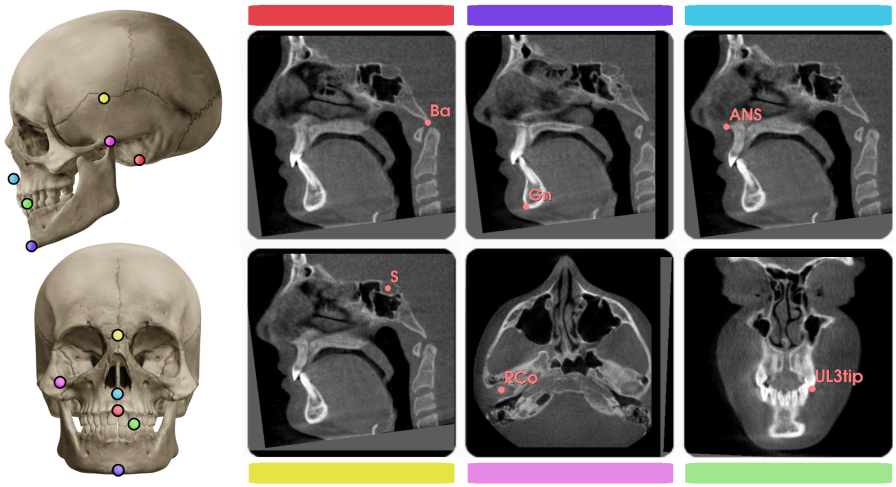

Figure 1: Selection of 6 different landmark, 2 in each bone group (Cranial base, Mandible and Maxilla)

3. Related work

Scanning-based Systems: A majority of detection solutions are scanning-based systems, especially when working in 3D data. A patch of local volumetric intensity boxes are extracted from an image, and a classifier then learn to distinguish the appearance of the target object from the rest of the sampled anatomy. Machine learning models like probabilistic boosting trees (Tu, 2005) are usually used for this. Other deep learning approaches use convolutional neural networks, sparse adaptive deep neural networks (Ghesu et al., 2016) and/or 3D Unet (Çiçek et al., 2016). However, the major drawback of these approaches is that at prediction time the model must scan the entire space of a given image set. With images of up to $600 \times 600 \times 600$ voxels, the memory usage grows exponentially, and so does the computation time.

Regression-based Systems: Regression-based systems learn relative displacement vectors pointing at the landmark location. These systems have been observed to use use random regression forests (Donner et al., 2013), random-ferns (Pauly et al., 2011), and deep convolutional neural networks (Erhan et al., 2014). These solutions highly improve the scanning-based systems' efficiency but still lack robustness.

End-to-End Systems: Also called image-to-image systems, end-to-end system developers got their inspiration from the fully convolution network (FCN) architecture. They learn the mapping between original image and segmentation multi-masks (Long et al., 2015).

Atlas-based Systems: The localization tasks can also be solved using atlas-based registration (Fenchel et al., 2008), as well as multi-atlas-based registration methods (Isgum et al., 2009). However, once again, applying this to a set of large 3D images is memory and compute time-consuming.

Deep Reinforcement Learning (DRL) Systems: Ghesu et al. proposed a method to solve most of the previously mentioned constraints using the capabilities of deep reinforcement learning and multi-scale image analysis (Ghesu et al., 2017). An artificial agent is trained to distinguish the target anatomical object from the rest of the body while learning how to navigate to this object in the volumetric space. DRL can be complicated to implement with the combination of a target network and the use of memory replay. The next section will describe how we tried to simplify the training task.

4. Methods

Our work relies on two principles: a multi-scale environment, and a search agent inspired by the behavioral problem solved as described in Deep Reinforcement Learning Systems. In this paper the behavior classification is solved using imitation learning. This approach is easier to implement and to train. It allows to use deeper neural networks that encode a wider range of image features.

4.1. Environment

A prerequisite to the landmark detection is data preprocessing. Twenty of the 60 CBCTs were acquired at a resolution of a 0.3mm, which was the highest resolution available. All the volumes were re-sampled to an isotropic resolution of 0.3mm for the finest scale level. We want the agent to learn different scales of the structures of interest. For our multi scale-space we used an additional low-resolution level at an isotropic spatial resolution of 1mm. The image histograms were also re-scaled to have a better contrast and the data was normalized to a [-1.0, 1.0] interval (Figure 2).
For each CBCT, 32 landmark positions were marked by clinicians and stored as a fidcuial list. For the training, the 6 selected landmark positions are mapped to the discrete image coordinates and stored in the environment memory.

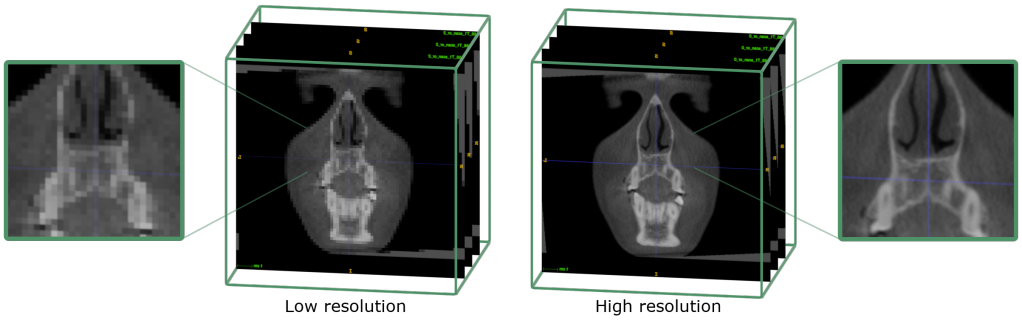

Figure 2: Visualization of the environment. On the left, the low 1mm resolution was re-scaled from the high-resolution 0.3mm scan on the right.

4.2. Agent

The protagonist of this work is the agent. It's a virtual object whose goal is to reach its target position (the landmark) by moving inside an environment. With a set of 6 possible actions, it is able to move move from one voxel to another by going up, down, left, right, front or back. It has a 3D field of view (FOV) as shown in Figure 3, a $Lx \times Ly \times Lz$ crop inside the environment around the agent position. A 0 padding is applied to the part of the FOV that is outside the scan. The size of the FOV is an important parameter, and we have to make sure that enough relevant image features can be extracted at the current location while limiting memory usage. This crop will be the "state" of the agent.
Our agents are made of deep networks for feature extraction (FeatNet), followed by fully connected non-linear function approximation layers. With significant improvements in results for image parsing tasks, deep learning systems are now an important part of the innovation in the field of machine learning. In this paper we compared the accuracy of two different 3D FeatNets, a Densely connected convolution network (DenseNet) (Huang et al., 2017), and a Deep residual network (ResNet) (Boroumand et al., 2018). The FeatNet is made of convolution layers which are trained to capture discriminative image features. It takes as input the state, and outputs a vector describing image information. This vector is then fed into the fully connected dense layers that output a probability vector $P \in \mathbb{R}^6$ of the best action to take from the input state. The agent moves following the highest probability.

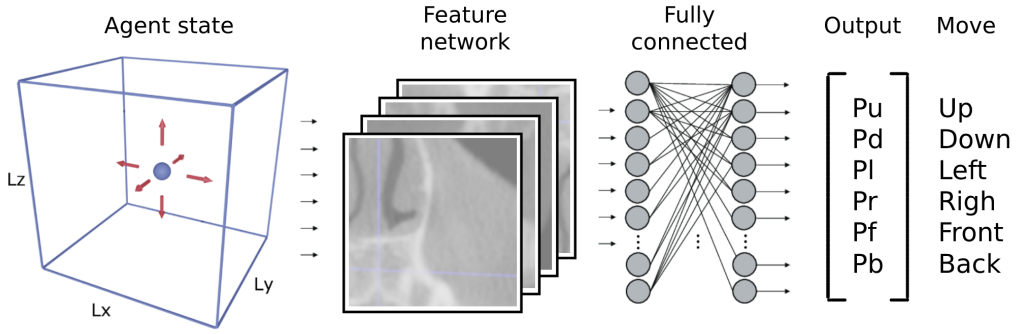

Figure 3: Visualization of the agent's $Lx \times Ly \times Lz$ field of view (blue box), and the 6 possible moves (red arrows) after the network prediction.

4.3. Training the agents

Our data was split by patients, 70% (42) for the training, 10% (6) for the validation and 20% (12) for the test. An environment was generated for each patient, and the corresponding landmark position were loaded. One agent was created for each landmark, and their network weights were initialised using a Xavier uniform function. Each agent was trained using a list of tuples: [S,A] where S is the state, and A is the best action to take from that state. We chose A to be the action that minimizes the most the distance to the landmark among the 6 possible movements. The low-resolution and high-resolution scans had an average size of

$180 \times 180 \times 180$ and $600 \times 600 \times 600$ voxels respectively. It means that for each environment we had more than $200,000,000$ states that can be used to train the agent. To limit the memory usage we used the following strategy to generate the list of tuples for each agent:

- At the low-resolution level, we initialized K random position with a 20% chance to be within a radius R_{low} of a ground truth landmark (region where more precision is needed). The remaining 80% could be anywhere in the scan. The agent is supposed to find the landmark from any starting point at this level.
- At the high-resolution level, we initialized K random position within a radius R_{high} of a ground truth landmark, knowing that the agent should be in this radius after the search at the low-resolution.

The K positions at each level were generated evenly in the N environments selected for the training. At every training epoch, we updated r% of the K positions by new randomly selected ones. It's one of the most important part of our training strategy that allowed the agent to be trained in most of the scan region while reducing the memory usage. The agent had a different network for each scale. These networks were optimized using the pytorch library using a combination of back-propagation algorithm to compute the network gradients (in this work, the cross entropy loss) and an ADAM optimizer. All the steps are summarized in Algorithm 1. The training was done on an NVIDIA Quadro RTX 6000/8000 GPU with a batch size of 100, $Lx = Ly = Lz = 64$, $K = 10,000$, $N = 2$, $r = 50\%$ and $R_{low} = R_{high} = 30$ voxels and it took about 2h for one agent to be trained and reach a good accuracy.

4.4. Predict the landmarks position

To predict the landmark positions in a CBCT, we rescale it to the resolutions used during training (here 1mm and 0.3mm voxel size). For each landmark to predict, an agent is generated with its corresponding network. The landmark prediction is then made in 3 steps (Figure 4):

- **Step 1:** The prediction begins at the low-resolution level. The agent is placed in the middle of the scan to optimize the search time. Once the agent reaches a confident zone, it goes to the high-resolution layer.
- **Step 2:** The agent starts moving in the high-resolution from the confident zone. A preliminary estimation is set where the agent stops moving,
- **Step 3:** Now, a verification step is applied. This step consists of searching again in the high-resolution scan starting from 6 positions (in each direction) in a small radius from the predicted point in **Step 2** . The result is an average of the 6 predicted positions.

The stopping criteria is active at prediction time and is implemented using a visitation map. If the agent tries to reach a visited voxel it stops. The third step increased the prediction accuracy and compensated for a portion of the error caused by the discrete aspect of the space. All the steps are summarized in Algorithm 2.

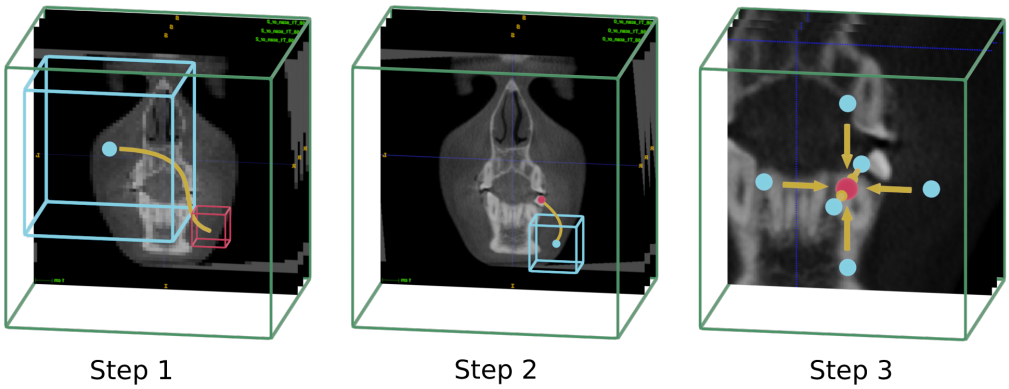

Figure 4: Visualization of the agent (blue) in the multi-scale environment (green) searching the target (red).

5. Results

Twelve patients were separated from the dataset to test the prediction accuracy. These images were never seen by the networks during training. We placed an agent for each of the 6 landmarks at the center of the low-resolution image and let them do the 3 searching steps. It takes around 4.2s on GPU for each landmark to predict. The prediction on the 12 scans required 8.8GB of cache memory and 2.1GB of GPU memory. Each agent did 90 moves on average to reach the landmark position.

A fiducial list that the clinician can use is generated with the predicted positions of the landmarks. To compute the prediction accuracy, we find the distance between each landmark in the ground truth (GT) fiducial list and the predicted one. After the position estimation in the low resolution, the agent was already found to be very close to the final position. Thus, it was sufficient to use only 2 resolutions for the prediction, making it faster while reducing the memory usage and without losing any accuracy. We compared the prediction results with different feature extraction strategies: DenseNet and a ResNet. The table DenseNet vs ResNet shows that depending on the landmark to search, one FeatNet was better than the other. On an average, the DenseNet showed better performance. Figure 5 shows the distribution of error in prediction (in mm) using DenseNet as the feature extraction for all landmarks. The more inconsistent accuracy of the UL3tip landmark compared to the Ba landmark can be explained by a strong variation from one patient to another. The tooth may be missing, or metal implants may create artifacts within the scan that make detection more difficult.

With an average error of 1.21mm we are below the clinician requested average error limit of 2mm. A prediction is considered to have failed when the agent is more than 5mm away from the ground truth. The 12 "test" images present different contrast and small variation in alignment. The detection can be considered to be robust with 0 failures out of the 72 predictions made.

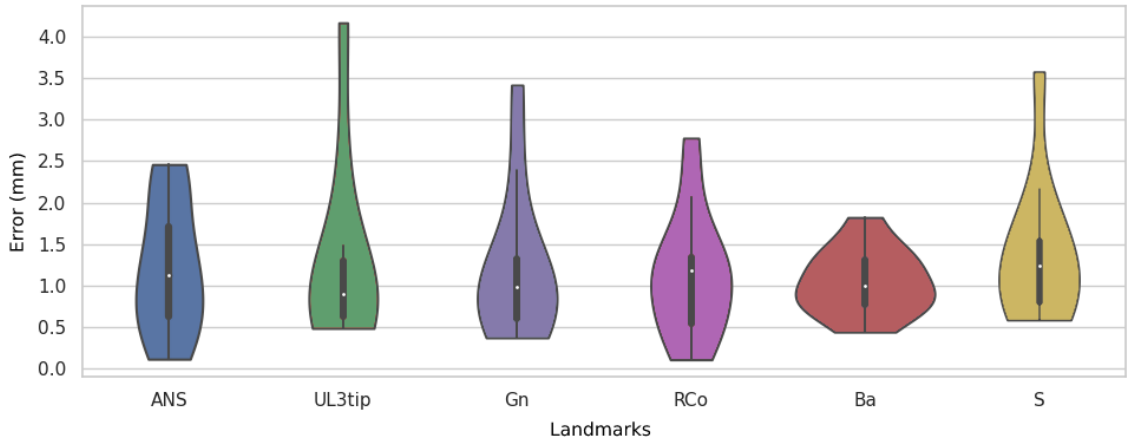

Figure 5: Violin plot of the distance in mm between the ground truth and the prediction for the 6 landmarks selected (using DenseNet).

6. Conclusion

This paper presents a novel method for robust and accurate anatomical landmarks localization for 3D medical imaging data. We combined the concept of scanning-based systems with smart displacement inside the scan using an agent. The training on a multi-resolution image enabled the artificial agent to systematically learn to find the targeted anatomical structures. Experiments show that our method is robust with no failed cases in the test set and accurate with less than 1.3 mm average error in landmark placement. The average detection speed of 4.2s is acceptable knowing the size of the high-resolution 3D-CBCT volumes used. Having separate models for each agent allows the clinician to make custom list of landmark to find. It also makes it easy for them to train new agents separately without compromising the previously trained models. Given the high robustness and good time performance, this method will be implemented in the open-source web based clinical decision support system (the Data Storage for Computation and Integration, DSCI) (Brosset et al., 2021), and in a user-friendly 3D Slicer module. For future work we will experiment to train a unique feature extraction layer common to all agents. The 28 remaining landmark will be trained and the method will be tested on another CT scan type. Downstream analysis, such as registration tasks or quantification (measuring distances, angles, etc.) will be performed.

Acknowledgments

This work was supported by NIDCR DE 024550, AA0F Dewel Memorial Biomedical Research award and by Research Enhancement Award Activity 141 from the University of the Pacific, Arthur A. Dugoni School of Dentistry.

References

Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018.

Serge Brosset, Maxime Dumont, Lucia Cevidanes, Reza Soroushmehr, Jonas Bianchi, Marcela L Gurgel, Romain Deleat-Besson, Celia Le, Antonio Ruellas, Marilia Yatabe, et al. Web infrastructure for data management, storage and computation. In *Medical Imaging 2021: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 11600, page 116001N. International Society for Optics and Photonics, 2021.

Juan R Cebral and Rainald Lohner. Efficient simulation of blood flow past complex endovascular devices using an adaptive embedding technique. *IEEE transactions on medical imaging*, 24(4):468–476, 2005.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

Antonio Criminisi, Jamie Shotton, Duncan Robertson, and Ender Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. In *International MICCAI Workshop on Medical Computer Vision*, pages 106–117. Springer, 2010.

Rémi Cuingnet, Raphael Prevost, David Lesage, Laurent D Cohen, Benoît Mory, and Roberto Ardon. Automatic detection and segmentation of kidneys in 3d ct images using random forests. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 66–74. Springer, 2012.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

Ren Donner, Bjoern H Menze, Horst Bischof, and Georg Langs. Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization. *Medical image analysis*, 17(8):1304–1314, 2013.

Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, et al. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic resonance imaging*, 30(9):1323–1341, 2012.

Matthias Fenchel, Stefan Thesen, and Andreas Schilling. Automatic labeling of anatomical structures in mr fastview images using a statistical atlas. In *International Conference on*

Medical Image Computing and Computer-Assisted Intervention, pages 576–584. Springer, 2008.

Florin C Ghesu, Edward Krubasik, Bogdan Georgescu, Vivek Singh, Yefeng Zheng, Joachim Hornegger, and Dorin Comaniciu. Marginal space deep learning: efficient architecture for volumetric image parsing. *IEEE transactions on medical imaging*, 35(5):1217–1228, 2016.

Florin-Cristian Ghesu, Bogdan Georgescu, Yefeng Zheng, Sasa Grbic, Andreas Maier, Joachim Hornegger, and Dorin Comaniciu. Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):176–189, 2017.

Ben Glocker, Darko Zikic, and David R Haynor. Robust registration of longitudinal spine ct. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 251–258. Springer, 2014.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Ivana Isgum, Marius Staring, Annemarieke Rutten, Mathias Prokop, Max A Viergever, and Bram Van Ginneken. Multi-atlas-based segmentation with local decision fusion—application to cardiac and aortic segmentation in ct scans. *IEEE transactions on medical imaging*, 28(7):1000–1010, 2009.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

Marcel Lüthi, Christoph Jud, and Thomas Vetter. Using landmarks as a deformation prior for hybrid image registration. In *Joint Pattern Recognition Symposium*, pages 196–205. Springer, 2011.

Olivier Pauly, Ben Glocker, Antonio Criminisi, Diana Mateus, Axel Martinez Möller, Stephan Nekolla, and Nassir Navab. Fast multiple organ detection and localization in whole-body mr dixon sequences. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 239–247. Springer, 2011.

Christian Payer, Darko Štern, Horst Bischof, and Martin Urschler. Regressing heatmaps for multiple landmark localization using cnns. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 230–238. Springer, 2016.

Alison M Pouch, Hongzhi Wang, Manabu Takabe, Benjamin M Jackson, Joseph H Gorman III, Robert C Gorman, Paul A Yushkevich, and Chandra M Sehgal. Fully automatic segmentation of the mitral leaflets in 3d transesophageal echocardiographic images using multi-atlas joint label fusion and deformable medial modeling. *Medical image analysis*, 18(1):118–129, 2014.

Naseem Shah, Nikhil Bansal, and Ajay Logani. Recent advances in imaging technologies in dentistry. *World journal of radiology*, 6(10):794, 2014.

Darko Štern, Thomas Ebner, and Martin Urschler. From local to global random regression forests: exploring anatomical landmark localization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 221–229. Springer, 2016.

Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1589–1596. IEEE, 2005.

Lin Yang, Bogdan Georgescu, Yefeng Zheng, Yang Wang, Peter Meer, and Dorin Comaniciu. Prediction based collaborative trackers (pct): A robust and accurate approach toward 3d medical object tracking. *IEEE transactions on medical imaging*, 30(11):1921–1932, 2011.

JI Yu, JS Kim, HC Park, DH Lim, YY Han, HC Lim, and SW Paik. Evaluation of anatomical landmark position differences between respiration-gated mri and four-dimensional ct for radiation therapy in patients with hepatocellular carcinoma. *The British journal of radiology*, 86(1021):20120221–20120221, 2013.

Paul A Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C Gee, and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

Jun Zhang, Mingxia Liu, Le An, Yaozong Gao, and Dinggang Shen. Alzheimer's disease diagnosis using landmark-based features from longitudinal structural mr images. *IEEE journal of biomedical and health informatics*, 21(6):1607–1616, 2017.

Appendix A. DensNet vs ResNet

Table 1: DensNet vs ResNet error in mm

Bone group	landmark	FeatNet	Fail	Max	Mean+STD
Maxilla	ANS	DenseNet	0	2.45	1.21±0.75
		ResNet	0	1.88	1.14±0.43
	UL3tip	DenseNet	0	4.16	1.30±1.06
		ResNet	0	4.66	1.21±1.26
Mandible	Gn	DenseNet	0	3.41	1.20±0.85
		ResNet	0	2.43	1.17±0.62
	RCo	DenseNet	0	2.77	1.12±0.71
		ResNet	0	2.85	1.43±0.73
Cranial base	Ba	DenseNet	0	1.82	1.05±0.37
		ResNet	0	1.57	0.96±0.40
	S	DenseNet	0	3.57	1.29±0.78
		ResNet	0	2.45	1.31±0.51
Total	All	DenseNet	0	4.16	1.21±0.79
		ResNet	0	4.66	1.29±0.75

Appendix B. Algorithm

Algorithm 1: Training agents

Given N Environments: $E_1, E_2, \ldots, E_N$ with $d = 2$ resolutions
And 6 Agents: $A_1, A_2, \ldots, A_6$
Initialize number of starting points = K
for *all Agents* **do**
  for $i \leftarrow 1$ **to** d **do**
    Set FeatureNet
    Initialize DensLayerNet randomly using a Xavier uniform function.
  end
end
Initialize ratio of starting point to update $r = 50\%$
for *epoch* $\leftarrow 1$ **to** *number of epoch* **do**
  for *all Agents* **do**
    Generate dataset from the K starting points on each resolution
    Train network (FeatNet + DensLayerNet)
    Replace $K \times r$ random starting point for each resolution
  end
end

Algorithm 2: Searching landmarks

Given N high-resolution scans: $S_1, S_2, \ldots, S_N$
Generate N Environments from the scans: $E_1, E_2, \ldots, E_N$ with $d = 2$ resolutions
for *all landmark to search* **do**
    Create Agent
    Load search model for each resolution
end
for *all Environments* **do**
    for *all Agents* **do**
        Put the agent in the center of the low-resolution environment
        while *Agent is moving* **do**
            Get agent FOV
            Predict action to take and move
        end
        Change agent environment to the high-resolution
        while *Agent is moving* **do**
            Get agent FOV
            Predict action to take and move
        end
        Apply the verification step around the actual position
        Save the predicted landmark position in the environment
    end
    Export landmarks found as fiducial list
end