

VERSE: A Vertebrae Labelling and Segmentation Benchmark

Anjany Sekuboyina^{a,b}, Amirhossein Bayat^{a,b}, Malek E. Husseini^{a,b}, Maximilian Löffler^b, Markus Rempfler^c, Jan Kukačka^d, Giles Tetteh^a, Alexander Valentinitzsch^b, Christian Payer^e, Martin Urschler^f, Maodong Cheng^g, Dalong Cheng^g, Nikolas Lessmann^h, Yujin Huⁱ, Tianfu Wangⁱ, Dong Yang^j, Daguang Xu^j, Felix Ambellan^k, Stefan Zachow^k, Tao Jiang^l, Xinjun Ma^l, Christoph Angerman^m, Xin Wangⁿ, Qingyue Wei^o, Kevin Brown^p, Matthias Wolf^p, Alexandre Kirschnerberg^q, Elodie Puybareau^q, Björn H. Menze^{☆a}, Jan S. Kirschke^{☆b}

^aDepartment of Informatics, Technical University of Munich, Germany.

^bDepartment of Neuroradiology, Klinikum Rechts der Isar, Germany.

^cFriedrich Miescher Institute for Biomedical Engineering, Switzerland

^dInstitute of Biological and Medical Imaging, Helmholtz Zentrum München, Germany

^eInstitute of Computer Graphics and Vision, Graz University of Technology, Austria

^fUniversity of Auckland, New Zealand

^gComputer Vision Group, iFLYTEK Research South China, China

^hDepartment of Radiology and Nuclear Medicine, Radboud University Medical Center Nijmegen, The Netherlands

ⁱHealth Science Center, Shenzhen University, China

^jNVIDIA Corporation, USA

^kZuse Institute Berlin, Germany

^lDamo Academy, Alibaba Group, China

^mDepartment of Mathematics, University of Innsbruck, Austria

ⁿDepartment of Electronic Engineering, Fudan University, China

^oDepartment of Radiology, University of North Carolina at Chapel Hill, USA

^pSiemens Healthineer, USA

^qEPITA Research and Development Laboratory (LRDE), Le Kremlin-Bicêtre, France

Abstract

In this paper we report the challenge set-up and results of the Large Scale Vertebrae Segmentation Challenge (VERSE) organized in conjunction with the MICCAI 2019. The challenge consisted of two tasks, vertebrae labelling and vertebrae segmentation. For this a total of 160 multidetector CT scan cohort closely resembling clinical setting was prepared and was annotated at a voxel-level by a human-machine hybrid algorithm. In this paper we also present the annotation protocol and the algorithm that aided the medical experts in the annotation process. Eleven fully automated algorithms were benchmarked on this data with the best performing algorithm achieving a vertebrae identification rate of 95% and a Dice coefficient of 90%. VERSE'19 is an open-call challenge at its image data along with the annotations and evaluation tools will continue to be publicly accessible through its online portal.

Keywords: vertebrae detection and localization, spine segmentation, vertebrae labelling, vertebrae segmentation, computed tomography.

[☆]Equal contribution

Email address: anjany.sekuboyina@tum.de (Anjany Sekuboyina)

1. Introduction

The spine is an important part of the musculoskeletal system, sustaining and supporting the body and its organ structure while playing a major role in our mobility and load transfer. It also shields the spinal cord from injuries and mechanical shocks due to impacts. Efforts towards quantification and understanding of the biomechanics of the human spine involve quantitative imaging [1], finite element modelling (FEM) of the vertebrae [2], alignment analysis [3] of the spine and complex models [4]. Biomechanical alterations can cause severe pain and disability in the short term, but in the long term, they can be morbid, eg. osteoporosis leads to an 8-fold higher mortality rate [5]. In spite of their criticality, spinal deformities are popularly under-diagnosed [6, 7]. This calls for computer-aided assistance for an efficient and early detection of such deformities or diseases, enabling prevention or effective treatment. *Vertebral labelling* (also referred to as recognition, detection) and *vertebral segmentation* are two essential stages in understanding spine image data. Labelling and segmentation have numerous diagnostic consequences such as detecting and grading vertebral fractures, estimating the spinal curve, recognizing spinal deformities such as scoliosis and kyphosis etc. From a non-diagnostic perspective, these tasks enable more efficient biomechanical modelling, FEM analysis, surgical planning for metal insertions etc. Conventionally, computed tomography (CT) is a preferred modality to study the spine due to a high bone-soft tissue contrast. For a human, labelling the vertebrae is straightforward, except in cases with restricted field-of-view (FoV). But, segmenting them is unfeasible owing to the size of the problem. Annotating 25 objects-of-interest at voxel-level, with each object being of size $\sim 10^3$ voxels demand considerable effort. Automating these tasks also has numerous challenges: highly varying FoVs across datasets (unlike brain images), large scan sizes, highly correlating shapes of adjacent vertebrae, complex posterior morphology of the vertebrae, scan noise, scanner settings and multiple anomalies or pathologies being present. In particular, presence of vertebral fractures, metal implants, cement, or transitional vertebrae further prohibits generalizable automation.

Nonetheless, there exists a clinical necessity for an automatic, accurate, and robust spine processing algorithm. Over the recent years, automated spine image analysis has seen a growing attention (cf Fig. 1). Effectively all these approaches are *data-dependant*, i.e require annotated data to either learn from, or tune, or adapt parameters. However, they have either been validated on private datasets or on small public datasets. Consider SpineWeb¹, an archive for multi-modal spine data. It lists only two CT datasets: CSI2014 [8] and xVertSeg [9]. The former consists of 20 full-spine CT scans while the latter is a collection of 25 lumbar CT scans, both with voxel-level annotations and the latter for only the lumbar region. This is not surprising if one considers the annotation effort. Therefore, benchmarking the spine processing algorithms becomes difficult due to varying datasets, resulting in inconsistency in the reported results of the newly-proposed algorithms. In case of the publicly available data, drawing clinically reliable conclusions is ill-advised due to the small test sample.

¹spineweb.digitalimaginggroup.ca

Addressing the concerns of large scale data availability and providing a common benchmark for current algorithms has been the primary objective of the Large Scale Vertebrae Segmentation Benchmark (VERSE). We organized it as a challenge in conjunction with the international conference on Medical Image Computing and Computed Assisted Intervention (MICCAI) 2019. With VERSE'19, we released into public domain a diverse dataset of 160 spine multi-detector CT scans, the largest public spine CT dataset till date [10]. We then invited participants to benchmark their algorithms on the tasks of *vertebral labelling* and *vertebral segmentation*. In this paper, we present a detailed report of VERSE'19 in three key parts: (1) We introduce our in-house, semi-automated spine processing algorithm that enabled the medical experts to accurately annotate all 1735 vertebrae of the 160 CT scans in the VERSE'19 dataset, including the detailed annotation protocols, (2) We describe the robust evaluation and benchmarking process adopted to compare the VERSE'19 submissions, and (3) we present an overview of the participating entries including a detailed analysis of the individual results.

2. Configuring the VerSe Benchmark

The VERSE benchmark was organised as a competition in conjunction with the international conference of Medical Image Computing and Computer Assisted Intervention (MICCAI) 2019. This section describes the setup towards the challenge, introduces the the participating algorithms and the evaluation metrics employed. VERSE'19 is open-call and the data and its evaluation tools are available to the community for continual benchmarking at verse2019.grand-challenge.org.

2.1. Data Description

2.1.1. Multidetector CT Imaging

The imaging data concerning VERSE'19 consists of 160 CT imaging series of 141 patients. The data was collected across multiple multidetector CT scanners. Care was taken to compose the data such that it resembles a typical clinical distribution in terms of fields-of-view, scan setting, and findings in an emergency as well as in oncological and neurosurgical conditions. For example: it consists of a variety of FOVs including thorco-lumbar and cervico-thoraco-lumbar scans, a mix of sagittal and isotropic reformations, cases with vertebral fractures, metallic implants and foreign materials. Please refer to [10] for a detailed description of the dataset's selection criteria, its composition, and a clinical overview.

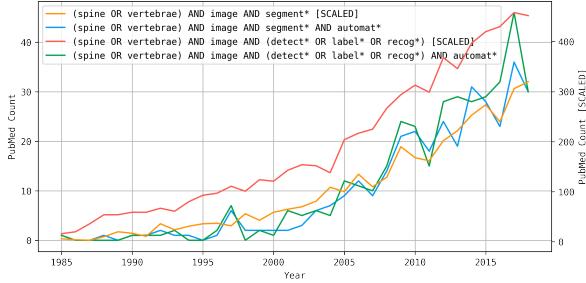


Figure 1: **Spine-related research on PubMed:** Plot indicating the number of published articles in spine imaging and automated spine image processing over the last three decades. Notice that automated processing algorithms have always formed only 10% of the total work dealing with spine processing

iii

2.1.2. Data Annotations: Protocol & Procedure

The data consists of two types of annotations: 1. 3D coordinate locations of the vertebral centroids for the *labelling* task and 2. voxel-level labels as segmentation masks for the *segmentation* task. Twenty five vertebrae (C1 to L6) were considered for annotation with labels from 1 to 25. Note that very few scans contained L6, in line with its rare occurrence in a population. For marking a *vertebral centroid*, annotators were asked to place the mark on the centre of mass of the vertebral body (viz. the region excluding the vertebral arch and processes). It should be noted that due to the special structure of C1, the centroid placed on its centre of mass physically manifests on the dens of C2. Note that only a minority of scans contained the full spine, implying that most scans included partially visible vertebrae at the top and bottom of the scan. Such *partially-visible* vertebrae were not labelled or segmented.

A human-machine hybrid annotation approach:

For annotating all 160 scans in the benchmark with more than 1725 vertebrae, a human-machine hybrid approach was employed to annotate the scans. Human experts were tasked with correcting the output of an automated algorithm as well as refining the corrections of other human raters. The centroids and the masks were manually and iteratively refined by one of four specifically trained medical students followed by further refinement, rejection or acceptance by one of the two trained radiologists with a joint experience of 22 years.

anduin0.1: A Spine Processing Framework. The interactive framework that aided the medical experts with reasonable initial annotations is referred to as the iBack framework. It splits the task in to three modules: 1. Spine detection, performed by a light-weight, fully-convolutional network predicting a low-resolution heatmap over the spine location using a fully-convolutional network, 2. Vertebra labelling, based on the Btrfly Net [11] architecture

working on sagittal and coronal maximum intensity projections (MIP) of the localized spine region, and finally, 3. Vertebral Segmentation, performed by an improved U-Net [12, 13] to segment vertebral patches, extracted at a high resolution, around the centroids predicted by the preceding stage. Fig. 2 gives a schematic of the entire framework. Note that the detection and labelling stages offer interaction, wherein

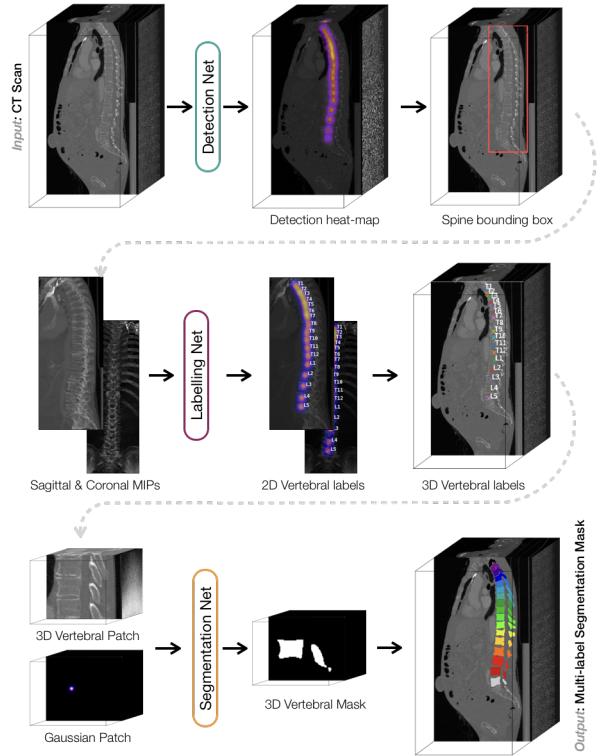


Figure 2: **Overview of *anduin0.1*:** A schematic of the semi-automated and interactive spine processing pipeline developed in-house. The **thick-black** lines indicate automated steps and the **dotted-grey** lines indicate an interactive step.

Table 1: List of the participating teams, the lead authors and the title of their algorithm as submitted in the technical report. Note that the team-wise colour codes are consistent throughout this work.

Team	Lead Author	Description
AlibabaDAMO	Jiang T.	SpineCenterSeg: A Keypoint-Based Instance Segmentation Framework for Vertebrae Segmentation
brown	Brown K.	Spine Segmentation and Registration
christian_payer	Payer C.	Vertebrae Localization and Segmentation with SpatialConfiguration-Net and U-Net [14]
christoph	Angermann C.	–
huyujin	Hu Y.	Large Scale Vertebrae Segmentation Using nnU-Net
iFLYTEK	Chen M.	An Automatic Multi-stage System for Vertebra Segmentation and Labelling
INIT	Wang X.	–
LRDE	Kirszenberg A.	–
nlessmann	Lessmann N.	Iterative fully convolutional neural networks
yangd05	Dong Y.	Vertebra Labeling and Segmentation in 3D CT using Deep Neural Networks
ZIB	Ambellan F.	Combining Template Matching with CNNs for Vertebra Segmentation and Identification

the user can alter the bounding box of the spine as well as the predicted vertebral centroids. Such *human-in-loop* design enabled collection of more accurate annotations with lesser human effort. Refer to Appendix A for a description of the network architecture, information on training and re-training schemes, as well as the post-processing steps at each stage.

2.2. The MICCAI 2019 Challenge

The first iteration of the VERSE benchmark was organised at MICCAI 2019 in Shenzhen, China. The 160 CT scans were split into a training set and two test sets with 80, 40, and 40 CT scans respectively. The second test set was hidden and inaccessible to public. Care was taken to preserve the composition across the data splits. The full training set (images, centroid annotations, and segmentation masks) was made publicly available in three phases over the summer of 2019 (till July) and submissions were solicited from the participants for the tasks of *labelling* and *segmentation*. Following this, the first phase of test data (only images, henceforth referred to as *Test-1*) was released on 7th August and participants were requested to submit the output of their algorithms on this data by e-mail to be considered for enrolment into the challenge. Alongside the predictions, participants were also asked to submit a technical report detailing their approach while cross-validating on the training data. Duration for *Test-1* was two weeks until 23rd August. Following this, over the next two weeks (until 6th September), the enrolled participants were asked to submit their code in a docker container for its evaluation on the hidden test data as part of the second test phase (*Test-2*). The rationale behind having a hidden test set was to prevent re-training of the algorithms on predictions from the test set.

2.2.1. Participating Methods.

Table 1 gives an overview of the teams that successfully registered and participated in the VERSE’19 benchmark. Altogether, 11 teams participated in at least one component of the challenge. The challenge contained four components: two phases (*Test-1* and *Test-2*), with each phase containing two tasks (*labelling* and *segmentation*). Therefore, we report four experiments for the benchmark. All the teams, except a few exceptions, were evaluated on all four components. The exceptions included: teams brown and huyujin participated only in the segmentation task, team brown did not make the docker submission, the docker containers of teams AlibabaDAMO and INIT were not sufficiently running during *Test-2*. For a detailed report on the methods adopted by each of the participating teams, we refer the reader to Appendix B.

2.2.2. Evaluation Metrics

Over the two tasks of labelling and segmentation, there exist twenty five objects of interest as vertebrae, as 3D coordinates for the former and segmentation masks for the latter. For evaluating the performance of the algorithms, we choose two metrics per task. Note that the metrics were chosen such that the algorithm will not be penalized if it labels or segments the partially-visible vertebrae in a scan.

Performance Measures for Labelling. As is established in the vertebral labelling literature, we evaluate the *Identification Rate* (*id.rate*) and localization distance (d_{mean}) for evaluating an algorithms labelling performance. Assuming a given scan contains N annotated vertebrae and denoting the true location of the i^{th} vertebra with \mathbf{x}_i and its predicted location with $\hat{\mathbf{x}}_i$, the vertebra i is correctly *identified* if $\hat{\mathbf{x}}_i$ is the closest landmark predicted to \mathbf{x}_i among $\{\mathbf{x}_j \forall j \text{ in } 1, 2, \dots, N\}$ and the Euclidean distance between the ground truth and the prediction is less than 20 mm, i.e $\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2 < 20$ mm. For a given scan, *id.rate* is then defined as the ratio of the correctly identified vertebrae to the total vertebrae present in the scan. Note that our definition of *id.rate* slightly deviates from its definition in [15], where *id.rate* is computed not at a scan-level but at a dataset level. Similarly, the localization distance is computed as $d_{\text{mean}} = \sum_{i=1}^N \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2$, the sum of the euclidean distances between the ground truth vertebral locations and their predictions.

Special cases: There will be cases where the prediction will contain more or fewer vertebrae than the ground truth. In the former case, the additional vertebral centroids are not considered for evaluation. However, when fewer vertebrae are predicted, d_{mean} is undefined as it is computed over every annotated centroid. Handling such missed vertebrae, we assign a maximum Euclidean distance of 1000 mm each missed prediction.

Performance Measures for Segmentation. For evaluating the segmentation task, we choose the ubiquitous Dice coefficient (Dice) and Hausdorff distance (*HD*). Denoting the ground truth by T and the algorithmic predictions by P , we evaluate both the metrics at a vertebrae level over all the vertebrae annotated in the ground truth. Dice score corresponding to the i^{th} vertebrae, denoted by $\text{Dice}(P_i, T_i)$ is computed as $2 \cdot |P_i \cap T_i| / |P_i| + |T_i|$, where $|\cdot|$ denotes the count of active voxels. At the scan level, vertebral Dice

scores are aggregated as $\text{Dice}(P, T) = (1/N) \sum_{i=1}^N \text{Dice}(P_i, T_i)$. Similarly, performance at a surface level is evaluated using Hausdorff distances. Denoting the surfaces of i^{th} vertebra by ∂P_i and ∂T_i and their surface points denoted by p_i and t_i , the Hausdorff distance between ∂P_i and ∂T_i is given by:

$$HD(\partial P_i, \partial T_i) = \max\{hd(\partial P_i, \partial T_i), hd(\partial T_i, \partial P_i)\},$$

where the directed Hausdorff distance is computed using all possible Euclidean distances between the points on the two surfaces as: $hd(\partial P_i, \partial T_i) = \sup_{p \in \partial P_i} \inf_{t \in \partial T_i} \|p - t\|_2$. $HD(P, T)$ is then computed as a mean over the vertebral surface distances. Note that HD is very sensitive to outlying voxels in the mask. To counter the effect of such noisy voxels, we compute HD over the largest connected component for every vertebral label.

Special cases: As with d_{mean} , HD is undefined if a ground truth vertebra is not segmented in the prediction. For such vertebrae, we assign a maximum Hausdorff distance of 100 mm before aggregating the distances over all the vertebrae in the scan.

2.2.3. Statistical Tests and Ranking

Inspired from [16] and [17], we compare the performance of the participating algorithms and rank them based on a scheme derived from a significance test. The value obtained from each scan in the cohort was treated a sample from a distribution and the Wilcoxon signed-rank test with a ‘greater’ or ‘less’ hypotheses testing (as appropriate for the performance metric) was employed to test the significance of the difference in performance between a pair of participants. A p -value of 0.001 was chosen as the threshold to ascertain a significant difference. Following this, a *point* was assigned to the better team. All possible such pairwise comparisons were performed for every performance measure, i.e for *id.rate* and d_{mean} for the labelling task and for Dice and HD for the segmentation tasks. Each comparison awards a point to a certain team unless the difference is not statistically significant. For every measure, the points are aggregated at a team level and normalized with the total number of participating teams in the experiment to obtain a score between 0 and 1. Lastly, the for every team, the normalized points across the measures are combined as described in the next section.

3. Results

In this section, we report the performance measures of the participating algorithms in the *segmentation* and *labelling* tasks in Tables 2 and 3 respectively. Adjacent to these tables are the *points* scored by each of the team, computed as elaborated in the previous section. We also present the ensuing metric-wise point matrices and their binarized versions (thresholded at $p = 0.001$) in Figs. 4 and 5. Note that the performance is measured and reported for both the test phases: *Test-1* and *Test-2*.

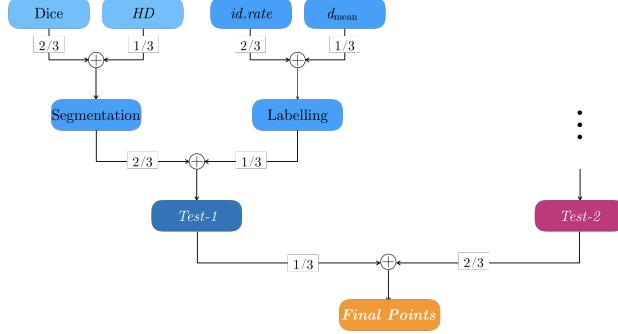


Figure 3: Protocol for obtaining the final ranking: Flow diagram of the weights assigned to each stage of evaluation in order to obtain the final ranks. Each stage represents the points obtained in said stage.

3.1. Final Ranking: Combining all the scores

VERSE‘19 is a collection of two tasks with two metrics each, evaluated over two phases. Fig. 3 illustrates how the performance of the algorithms over the multiple stages were combined to construct one ranking scheme. Table 4 reports the ranks thus obtained. The rationale of the organizers in choosing this scheme follows:

- d_{mean} and HD compared to $id.\text{rate}$ and Dice are weighted at a ratio of 1 : 2 in order to de-emphasize the contribution of the upper bounds chosen on the former measures in case of missing predictions.
- $Test-2$ has twice the weight as $Test-1$ as it was evaluated on completely hidden dataset, thus nullifying the chance of over-fitting or retraining on the test set.
- Lastly, the segmentation task has twice the weight of the labelling task as the latter can possibly be a consequence of the former.

4. Conclusions

In this paper we presented a report on the Large Scale Vertebra Segmentation Challenge (VERSE‘19) consisting of the vertebrae labelling and segmentation tasks. For this we prepared a largest spine dataset with accurate voxel-level annotations. We elaborate the algorithm used for generating said annotations while also summarizing the algorithms the participated in the challenge. The best performing algorithm achieved a Dice coefficient of 89.8% and an vertebral identification rate of 94.2% on a hidden test set.

Table 2: **Segmentation**: Performance of various teams (in alphabetical order) in the segmentation task for the three sets of predictions. Note: brown (no docker submission), AlibabaDAMO and INIT (erroneous docker) have missing numbers.

Team	Test-1		Test-2		Team	Test-1		Test-2	
	Dice	HD	Dice	HD		Dice	HD	Dice	HD
AlibabaDAMO	82.70	11.22	—	—	AlibabaDAMO	4	4	—	—
brown	62.69	35.90	—	—	brown	1	1	—	—
christian_payer	90.90	6.35	89.80	7.34	christian_payer	8	8	5	5
christoph	43.14	44.27	46.40	42.85	christoph	1	2	0	1
huyujin	84.66	12.79	81.82	29.44	huyujin	4	4	3	3
iFLYTEK	93.01	6.39	83.74	11.67	iFLYTEK	10	8	3	4
INIT	71.88	24.59	—	—	INIT	2	3	—	—
LRDE	13.97	77.48	35.64	64.52	LRDE	0	1	0	0
nlessmann	85.08	8.58	85.76	9.01	nlessmann	4	5	3	5
yangd05	76.74	14.09	67.06	28.76	yangd05	2	4	2	1
ZIB	67.02	17.35	68.96	19.25	ZIB	1	3	2	2

Table 3: **Labelling**: Performance of various teams (in alphabetical order) in the labelling task for the three sets of predictions. Note: brown (no docker submission, no label predictions), AlibabaDAMO, INIT (erroneous docker), and huyujin (no label predictions) have missing numbers.

Team	Test-1		Test-2		Team	Test-1		Test-2	
	id.rate	d_mean	Id.rate	d_mean		id.rate	d_mean	Id.rate	d_mean
AlibabaDAMO	89.82	7.39	—	—	AlibabaDAMO	3	5	—	—
brown	—	—	—	—	brown	—	—	—	—
christian_payer	95.65	4.27	94.25	4.80	christian_payer	3	7	3	5
christoph	55.80	44.92	54.85	19.83	christoph	1	1	1	1
huyujin	—	—	—	—	huyujin	—	—	—	—
iFLYTEK	96.94	4.43	86.73	7.13	iFLYTEK	5	7	2	4
INIT	84.02	12.40	—	—	INIT	2	3	—	—
LRDE	0.01	205.41	0.0	1000	LRDE	0	0	0	0
nlessmann	89.86	14.12	90.42	7.04	nlessmann	3	1	4	3
yangd05	62.56	18.52	67.21	15.82	yangd05	1	1	1	1
ZIB	71.63	11.09	73.32	13.61	ZIB	1	1	1	1

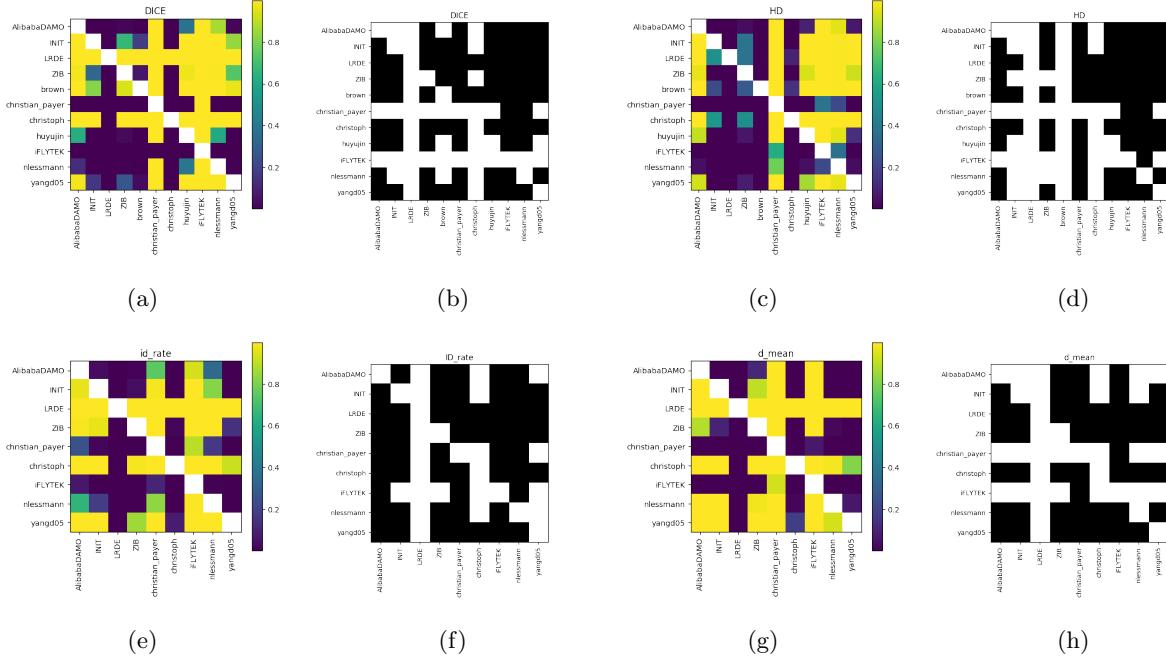


Figure 4: **Point matrices for Test-1:** Illustrating the p -value matrices and their binarized versions for every metric used. Top and bottom rows correspond to the segmentation and labelling tasks. Please find the metric corresponding to each matrix as the figure's title.

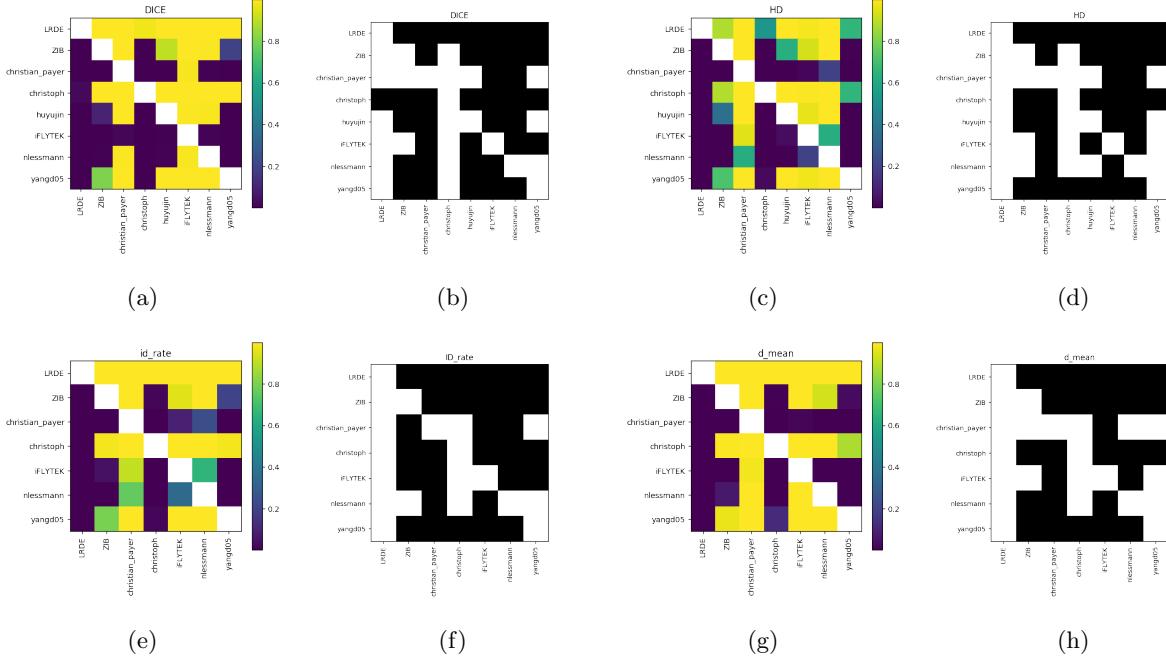


Figure 5: **Point matrices for Test-2:** Illustrating the p -value matrices and their binarized versions for every metric used. Top and bottom rows correspond to the segmentation and labelling tasks. Please find the metric corresponding to each matrix as the figure's title.

Table 4: **Final normalised point count**: Table indicates the final points obtained by each team according to the evaluation protocol described in this article. Maximum point value by a team can be 1.0.

Team	christian-payer	iFLYTEK	mlessmann	huyujin	yangd05	ZIB	AlibabaDAMO	christoph	INIT	brown	LRDE
Points	0.691	0.597	0.496	0.279	0.216	0.215	0.140	0.107	0.084	0.022	0.007

References

- [1] M. Löffler, N. Sollmann, K. Mei, A. Valentinitisch, P. Noël, J. Kirschke, T. Baum, X-ray-based quantitative osteoporosis imaging at the spine, *Osteoporosis International*.
- [2] D. P. Anitha, T. Baum, J. S. Kirschke, K. Subburaj, Effect of the intervertebral disc on vertebral bone strength prediction: a finite-element study, *The Spine Journal*.
- [3] F. Laouissat, A. Sebaaly, M. Gehrchen, P. Roussouly, Classification of normal sagittal spine alignment: refounding the roussouly classification, *European spine journal : official publication of the European Spine Society, the European Spinal Deformity Society, and the European Section of the Cervical Spine Research Society* 27 (8).
- [4] T. R. Oxland, Fundamental biomechanics of the spine—what we have learned in the past 25 years and future directions, *Journal of Biomechanics* 49 (6).
- [5] J. A. Cauley, D. E. Thompson, K. C. Ensrud, J. C. Scott, D. Black, Risk of mortality following clinical fractures, *Osteoporosis International* 11 (7).
- [6] D. Müller, J. S. Bauer, M. Zeile, E. J. Rummery, T. M. Link, Significance of sagittal reformations in routine thoracic and abdominal multislice ct studies for detecting osteoporotic fractures and other spine abnormalities, *European Radiology* 18 (8).
- [7] A. L. Williams, A. Al-Busaidi, P. J. Sparrow, J. E. Adams, R. W. Whitehouse, Under-reporting of osteoporotic vertebral fractures on computed tomography, *European Journal of Radiology* 69.
- [8] J. Yao, J. E. Burns, H. Munoz, R. M. Summers, [Detection of vertebral body fractures based on cortical shell unwrapping](#), *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* 15.
URL csi-workshop.weebly.com/challenges.html
- [9] R. Korez, B. Ibragimov, B. Likar, F. Pernuš, T. Vrtovec, [A framework for automated spine and vertebrae interpolation-based detection and model-based segmentation](#), *IEEE Transactions on Medical Imaging*

Imaging 34 (8).

URL lit.fe.uni-lj.si/xVertSeg/

- [10] M. Löffler, A. Sekuboyina, A. Jakob, A. Grau, A. Scharr, M. Husseini, M. Herbell, C. Zimmer, T. Baum, J. Kirschke, A vertebral segmentation dataset with fracture grading, Radiology: Artificial Intelligence Under Review.
- [11] A. Sekuboyina, M. Rempfler, J. Kukačka, G. Tetteh, A. Valentinitisch, J. S. Kirschke, B. H. Menze, Btrfly net: Vertebrae labelling with energy-based adversarial learning of local spine prior, in: Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, 2018.
- [12] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 2015.
- [13] A. G. Roy, N. Navab, C. Wachinger, Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks, in: Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, 2018.
- [14] C. Payer, D. Štern, H. Bischof, M. Urschler, Coarse to fine vertebrae localization and segmentation with spatialconfiguration-net and u-net, in: Proceedings of the International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), 2020, p. In Press.
- [15] B. Glockner, J. Feulner, A. Criminisi, D. R. Haynor, E. Konukoglu, Automatic localization and identification of vertebrae in arbitrary field-of-view ct scans, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012, 2012.
- [16] L. Maier-Hein, M. Eisenmann, A. Reinke, S. Onogur, M. Stankovic, et al., Why rankings of biomedical image analysis competitions should be interpreted with care, Nature Communications.
- [17] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, et al., The multimodal brain tumor image segmentation benchmark (brats), IEEE Transactions on Medical Imaging.
- [18] A. Sekuboyina, M. Rempfler, A. Valentinitisch, B. H. Menze, J. S. Kirschke, Labelling vertebrae with 2d reformations of multidetector ct images: An adversarial approach for incorporating prior knowledge of spine anatomy, Radiology: Artificial Intelligence (2020) In Press.

Appendix A. Description of *anduin0.1*

Given the CT scan of a spine, our framework aims to predict accurate voxel-level segmentation of the vertebrae by split the task in to three sub-tasks: spine detection, vertebrae labelling, and vertebrae segmentation. In the following section, the network architectures, loss functions, and training and inference details of each of these modules is elaborated. Fig. 2 gives an overview of the proposed framework and Fig. A.6 details the architectures of the networks employed in the three sub-tasks.

Appendix A.1. Notation .

The input CT scan is denoted by $\mathbf{x} \in \mathbb{R}^{h \times w \times d}$ where h , w , and d are the height, width, and depth of the scan respectively. The annotations available to us are, (1) the vertebral centroids, denoted by $\{\mu_i \in \mathbb{R}^3\}$ for $i \in \{1, 2, \dots, N\}$. These are used to construct the ground truth for the detection and labelling tasks, denoted by \mathbf{y}_d and \mathbf{y}_l , respectively. (2) the multi-label segmentation masks, denoted by $\mathbf{y}_s \in \mathbb{Z}^{h \times w \times d}$.

Appendix A.2. Spine Detection

For detecting the spine, we propose a parametrically-light, 3D, fully convolutional network operating at an isotropic resolution of 4 mm. This network regresses a 3D volume consisting of Gaussians at the vertebral locations as shown in Fig. A.6. The Gaussian heatmap is generated at a resolution 1 mm with a standard deviation, $\sigma = 8$, and then downsampled to a resolution of 4 mm. Additionally, spatial squeeze and channel excite blocks (SSCE) are employed to increase the network’s performance-to-parameters ratio. Specifically, the probability of each voxel being a *spine voxel* or a *non-spine* one is predicted by optimizing a combination of ℓ_2 and binary cross-entropy losses as shown:

$$\mathcal{L}_{\text{detect}} = \|\mathbf{y}_d - \tilde{\mathbf{y}}_d\|_2 - H(\sigma(\mathbf{y}_d), \sigma(\tilde{\mathbf{y}}_d)) \quad (\text{A.1})$$

where \mathbf{y}_d is constructed by concatenating the Gaussian location map with a background channel obtained by subtracting the foreground from 1, $\tilde{\mathbf{y}}_d$ denotes the prediction of whose foreground channel represents the desired location map, and $\sigma(\cdot)$ and $H(\cdot)$ denote the softmax and cross-entropy functions.

Appendix A.3. Stage 2: Vertebrae Labelling

For labelling the vertebrae, we adapt and improve the Btrfly net [11, 18] that works on two-dimensional sagittal and coronal maximum intensity projections (MIP). By virtue of the spine’s extant obtained from the previous component, MIPs can now be extracted from a region focused on the spine, thus eliminating occlusions from ribs and pelvic bones. Cropping the scans to the spine region also makes the input to the labelling stage more uniform, thus improving the training stability. The labelling module works at 2 mm isotropic resolution and is trained by optimizing the loss function that is a combination of the sagittal and coronal components, $\mathcal{L}_{\text{label}} = \mathcal{L}_{\text{label}}^{\text{sag}} + \mathcal{L}_{\text{label}}^{\text{cor}}$, where the loss of each view is given by:

$$\mathcal{L}_{\text{label}}^{\text{sag}} = \|\mathbf{y}_l^{\text{sag}} - \tilde{\mathbf{y}}_l^{\text{sag}}\|_2 + H(\sigma(\mathbf{y}_l^{\text{sag}}), \sigma(\tilde{\mathbf{y}}_l^{\text{sag}})), \quad (\text{A.2})$$

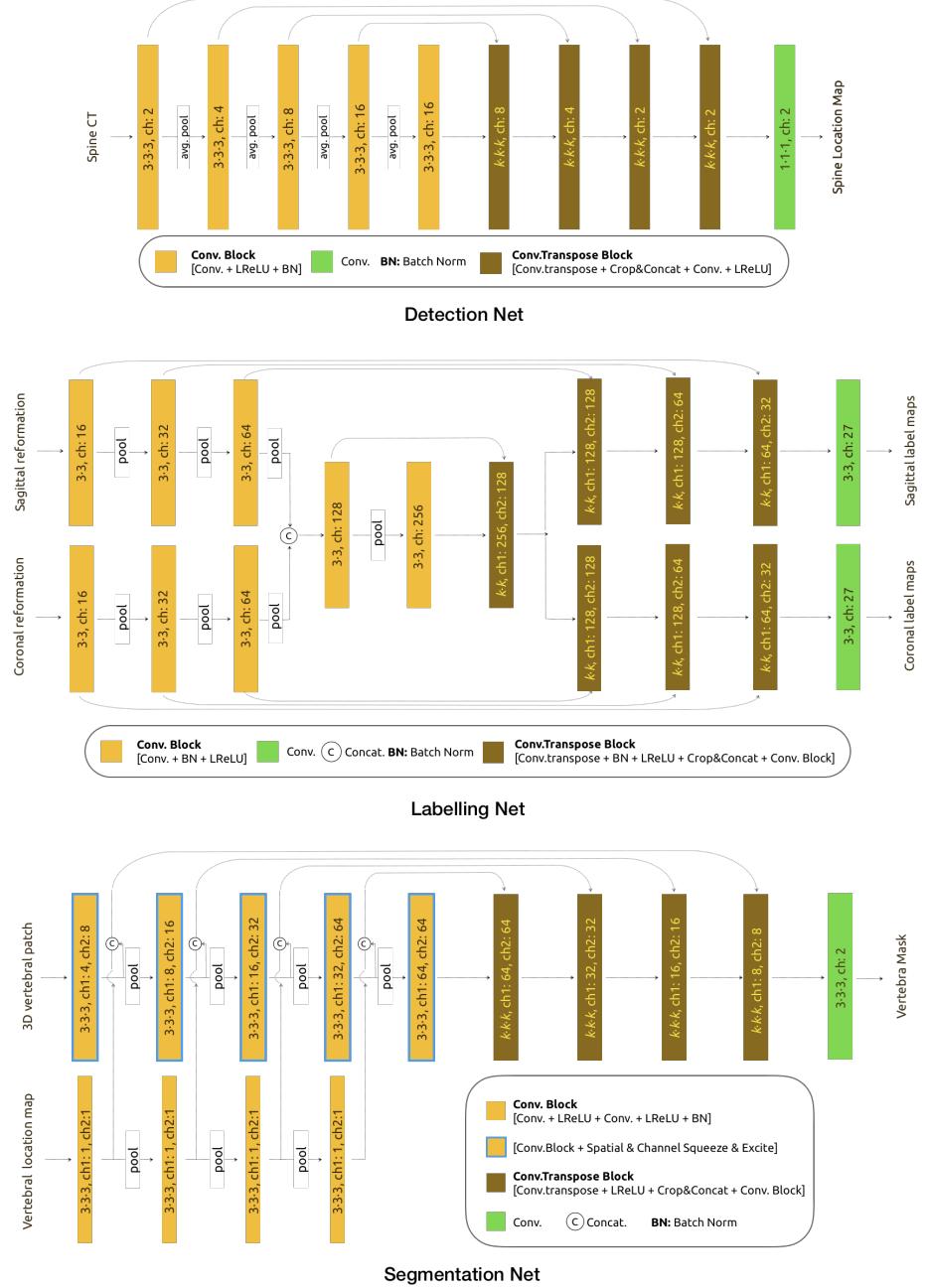


Figure A.6: **Architectures:** Detailed network architectures of the three stages in *anduin0.1*: the spine detection, vertebrae labelling, and the vertebra segmentation stages.

where \tilde{y}_l^{sag} is the prediction of the net's sagittal-arm of the Btrfly net and ω denotes the median frequency weight map giving a higher weight to the loss originating from less frequent vertebral classes.

Appendix A.4. Stage 3: Vertebral Segmentation

Once the vertebrae are labelled, their segmentation is posed as a binary segmentation problem. This is done by extracting a patch around each vertebral centroid predicted in the earlier stage and segmenting the

vertebra of interest. An architecture based on the U-Net working at a resolution of 1 mm is employed for this task. Additionally, SSCE blocks are incorporated after every convolution and upconvolution blocks. Importantly, as there will be more than one vertebra within a patch, a vertebra-of-interest (VOI) arm is used to point the segmentation network to delineate the vertebra of interest. The VOI arm is an encoder parallel to the image encoder as shown in Fig. A.6, processing a 3D Gaussian heatmap centred at the vertebral location predicted by the labelling stage. The feature maps of the VOI arm are concatenated to those of the image encoder at every resolution. The segmentation network is trained using a standard binary cross-entropy as a loss.

Algorithm 1: Pseudocode for inference on *anduin0.1*

Input: \mathbf{x} , a 3D MDCT spine scan

Output: Vertebral centroids & segmentation masks

DETECTION

- 1 $\mathbf{x}_d = \text{resample_to_4mm}(\mathbf{x})$
- 2 $\mathbf{y}_d = \text{predict_spine_heatmap}(\mathbf{x}_d)$
- 3 $bb = \text{construct_bounding_box}(\mathbf{y}_d, \text{threshold} = T_d)$
- 4 Interaction: Alter bb by *mouse-drag* action.

LABELLING

- 5 $\mathbf{x}_l = \text{resample_to_2mm}(\mathbf{x})$
- 6 $bb = \text{upsample_bounding_box}(bb, \text{from} = 4\text{mm}, \text{to} = 2\text{mm})$
- 7 $\mathbf{x}_{sag}, \mathbf{x}_{cor} = \text{get_localised_mips}(\mathbf{x}_l, bb)$
- 8 $\mathbf{y}_{sag}, \mathbf{y}_{cor} = \text{predict_vertebral_heatmaps}(\mathbf{x}_{sag}, \mathbf{x}_{cor})$
- 9 $\mathbf{y}_l = \text{get_outer_product}(\mathbf{y}_{sag}, \mathbf{y}_{cor})$
- 10 $\text{centroids} = \text{heatmap_to_3D_coordinates}(\mathbf{y}_l, \text{threshold} = T_l)$
- 11 Interaction: Insert missing vertebrae, delete spurious predictions, drag incorrect predictions.

SEGMENTATION

- 12 $\mathbf{x}_s = \text{resample_to_1mm}(\mathbf{x}); mask = \text{np.zeros_like}(\mathbf{x}_s)$
- 13 **for** *every centroid in centroids* **do**
- 14 $p = \text{get_3D_vertebral_patch}(\mathbf{x}_s, \text{centroid})$
- 15 $p_{mask} = \text{binary_segment_vertebra_of_interest}(p)$
- 16 $p_{mask} = \text{index_of}(mask, \text{centroid}) * p_{mask}$
- 17 $mask = \text{put_vertebrae_in_mask}(p_{mask})$
- 18 **end**

Appendix A.5. Inference & Interaction

Simplifying the flow of control throughout the pipeline, Algo. 1 describes the inference routine given a spine CT scans and various points where medical experts can interact with the results, thus improving its overall performance.

Appendix B. Summary of the Participating Algorithms

In this section, we include the technical reports submitted by the participating algorithms.

1. ■ AlibabaDAMO: Refer to page [xvii](#)
2. ■ brown: Refer to page [xx](#)
3. ■ christian_payer: Refer to page [xxii](#)
4. ■ christoph: Refer to page [xxvi](#)
5. ■ huyujin: Refer to page [xxxii](#)
6. ■ iFLYTEK: Refer to page [xxxiii](#)
7. ■ INIT: Refer to page [xl](#)
8. ■ LRDE: Refer to page [xliv](#)
9. ■ nlessmann: Refer to page [lii](#)
10. ■ yangd05: Refer to page [lv](#)
11. ■ ZIB: Refer to page [lvii](#)

SpineCenterSeg: A Keypoint-Based Instance Segmentation Framework for Vertebrae Segmentation

Tao Jiang¹ and Xinjun Ma¹

Damo Academy, Alibaba Group, China

jiangtao.jiangtao@alibaba-inc.com, xinjun.mxj@alibaba-inc.com

1 Our Algorithm Framework

The proposed keypoint-based instance segmentation framework for vertebrae segmentation consists of three components: (a) A binary segmentation network (b) A position prediction network (c) A vertebrae labeler. The overall flowchart is given as Fig. 1. In short, we decompose the instance segmentation problem into several sub-problems so that we can easily train a deep neural network for each of them. Firstly, we use a deep neural network to separate vertebral column from background and other tissues without differentiating each vertebrae. The output of this segmentation network, which is a binary volume, is further fed into the position prediction network. The position prediction network has two heads, one of which is landmark detection branch outputting the 3D heat-map of the predicted vertebrae center point [1]. The other head of the position prediction network will estimate a 3D vector field, in which the individual vector of each pixel points to its corresponding vertebrae center. At last, with the predicated 3D vector field and center points, a max likelihood estimation is performed to extract each vertebrae.

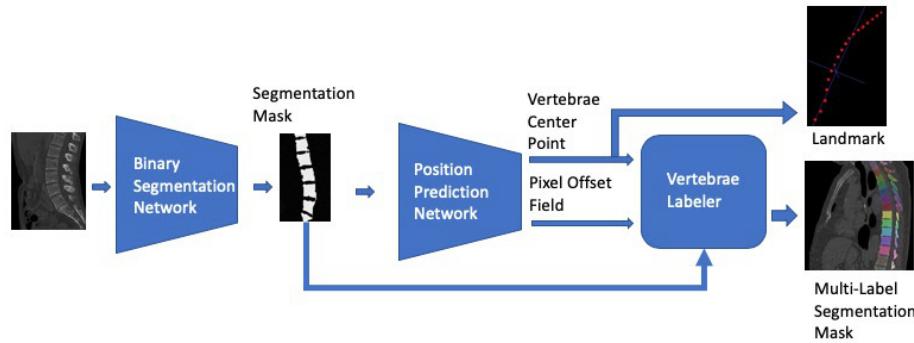


Fig. 1. The Overall flowchart of our keypoint-based instance segmentation framework.

1.1 Binary Segmentation Network

The backbone of our binary segmentation network is stacked hourglass as [2]. Our segmentation network takes input as several consecutive sagittal views and predicts the binary mask of the central sagittal view. At inference time, we also put the same number of sagittal views into input tensor but only predicts the central view.

1.2 Position Prediction network

The backbone of our position prediction network is FPN as [3] but with 3D convolution. Two separate heads will predict 3D heatmap of vertebral centers and offset vector field relative to the corresponding vertebral centers as in Fig. 2. The input of the network is cropped 3D patches from original CT volume. After obtaining 3D heatmap, a max pooling is performed to get peak points, corresponding to vertebral center points, and with the offset vector field we assign each foreground pixel to its closest center point. Using offset and keypoint prediction, each foreground pixel will have a unique vertebral label and thus a instance segmentation mask is obtained.

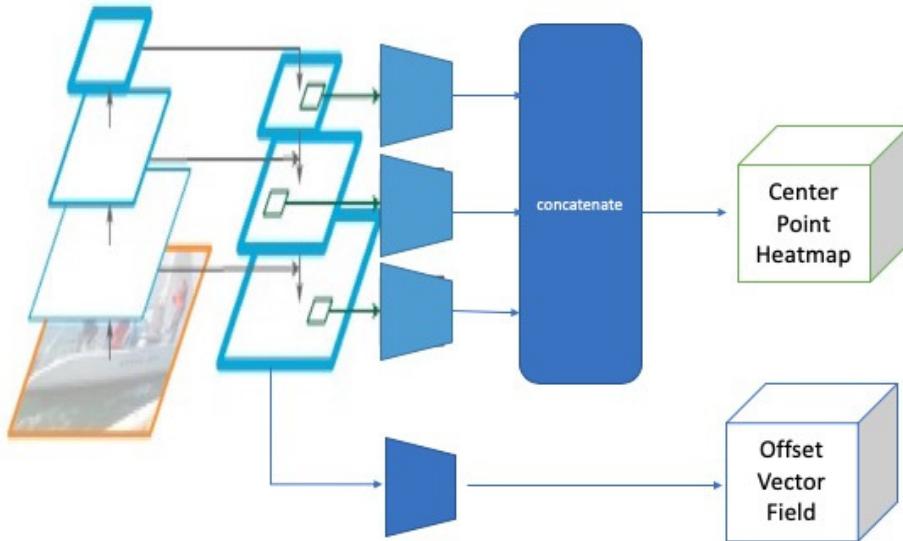


Fig. 2. Position Prediction Network

1.3 Vertebrae Labeler

Finally, each vertebrae subvolume is cropped out according to the instance segmentation result, and these subvolumes form the input tensor of a recurrent

neural network, which learns correct classification for each vertebrae. Combining classification tags, instance segmentation mask and keypoint prediction we obtain the final result as shown in Fig. 3.

As presented in Fig. 3 and Table. 1, our novel framework is able to achieve competitive result on the phase 1 test set in Verse2019 challenge.

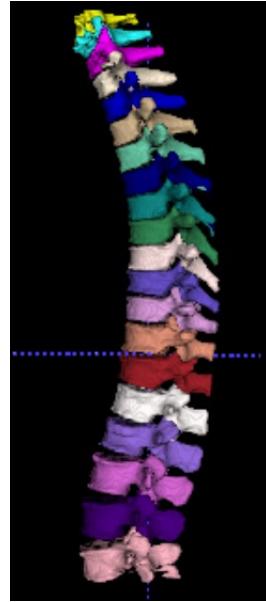


Fig. 3. Segmentation Result

Table 1. Metrics.

Score Position	MLD	Dice	Precision	Recall	ID Rate	Hausdorff Distance
0.9205	7.1985	0.8178	0.8232	0.9205	0.8787	29.7313

References

1. Xingyi Zhou, Dequan Wang, Philipp Krhenbhl: Objects as Points. <https://arxiv.org/abs/1904.07850>
2. Alejandro Newell, Kaiyu Yang, Jia Deng: Stacked Hourglass Networks for Human Pose Estimation. In: ECCV 2016. https://doi.org/10.1007/978-3-319-46484-8_29
3. Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, Serge J. Belongie: Feature Pyramid Networks for Object Detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 936-944. (2016)

Spine Segmentation and Registration*

Kevin Brown^{1,2} and Matthias Wolf¹

¹ Siemens Healthineers, Malvern PA 19355, USA

² New York University, New York, New York 10009 kab695@nyu.edu

Abstract. We aim to segment individual vertebrae in the provided CT images. A bounding box around each vertebrae is computed, and a residual U-net is used to segment the box, once registered to a common space.

Keywords: Spine Segmentation · Spine Labeling · Deep Learning.

1 Method

1.1 Objective function

We used a dice objective function because of its previous success in accurate classification near segmentation boundaries. The dice coefficient measures the degree of overlap between two sets. For two binary sets ground truth (G) and predicted class membership (\hat{G}) with (N) elements each, the dice coefficient can be written as

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \quad (1)$$

where each p_i and g_i are binary labels. We set p_i in $[0, 1]$ from the softmax layer representing the probability that the i^{th} voxel is in the foreground class. Each g_i is obtained from a one-hot encoding of the ground-truth labeled volume of tissue class. For model evaluation, class labels were always assigned binary labels from the most likely class per voxel.

1.2 False Positive, False Negative

We added a weighted false-positive/false-negative loss term to provide smoother convergence:

$$L_{FNFP} = \sum_{i \in I} w_i p_i (1 - g_i) + \sum_{i \in I} w_i (1 - p_i) g_i \quad (2)$$

$$w_i = \gamma_e \exp(-d_i^2 / \sigma) + \gamma_c f_i \quad (3)$$

* Supported by Siemens Healthineers.

Where d_i is the euclidean distance to the nearest class boundary, f_i is the frequency of the ground truth class at voxel i . Here, σ is chosen to be 10 voxels, and γ_e and γ_c are 5 and 2, respectively.

Our final loss was a weighted combination of the weighted loss above and the dice coefficient:

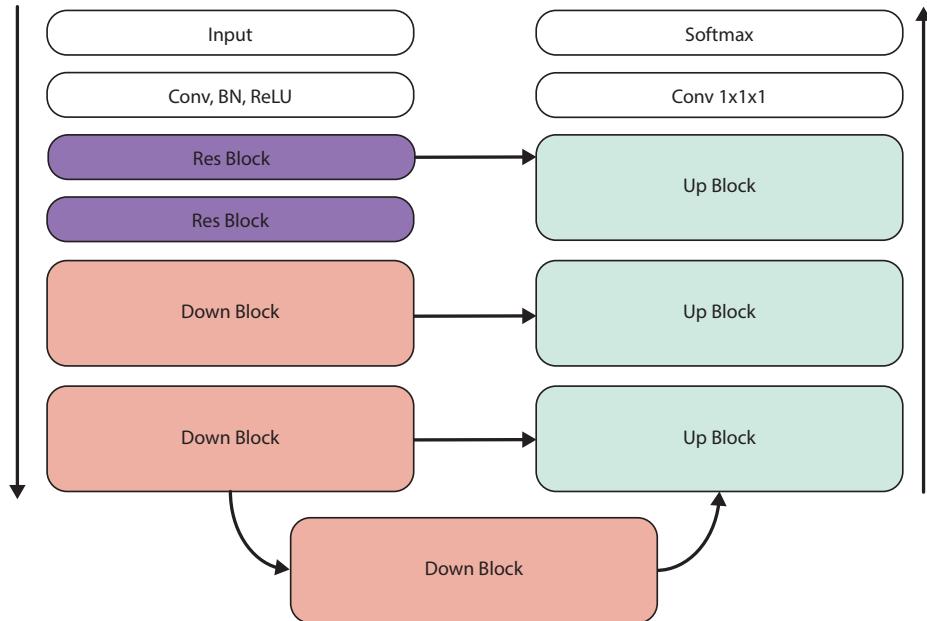
$$L = L_{FNFP} + \alpha L_D \quad (4)$$

α was chosen to be 0.5 and incrementally lowered throughout training.

1.3 Model structure

We employed a residual U-net (Figure 1) with an input size of 64 by 64 by 64 voxels, and depth of 5 blocks. A bounding box around vertebra is identified via a regressed set of canonical landmarks. Each vertebra is then registered to a common 'atlas' space via these landmarks.

Fig. 1. Residual U-net



Vertebrae Localization and Segmentation with SpatialConfiguration-Net and U-Net

Christian Payer^{1,2[0000-0002-5558-9495]}, Darko Štern^{1,2[0000-0003-3449-5497]},
Horst Bischof^{1[0000-0002-9096-6671]}, and Martin Urschler^{2,3[0000-0001-5792-3971]}

¹Institute of Computer Graphics and Vision, Graz University of Technology, Austria

²Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria

³University of Auckland, New Zealand

1 Introduction

This technical report introduces our proposed pipeline for fully automatic vertebrae localization and segmentation in CT volumes for the VerSe 2019 Large Scale Vertebrae Segmentation Challenge. The challenge consists of two tasks, where the first one is to localize and label the centers of the individual vertebrae, and the second one is vertebrae segmentation. For more details of the dataset and creation of the annotations, visit the homepage of the challenge and see the respective publications [1, 4].

2 Method

We perform vertebrae localization and segmentation in a three-step approach. Firstly, due to the large variation of the field of view of the input CT volumes, a CNN with a coarse input resolution predicts the approximate location of the spine. Secondly, another CNN in higher resolution performs multiple landmark localization and identification of the individual vertebra centroids. Lastly, the segmentation CNN performs a binary segmentation of each located vertebra. The results of the individually segmented vertebrae are merged into the final multi-label segmentation.

2.1 Spine Localization

For localizing the approximate position of the spine, we use a variant of the U-Net [3] to regress a heatmap of the spinal centerline, i.e. the line passing through vertebral centroids, with an L2-loss [2]. The heatmap of the spinal centerline is generated by combining Gaussian heatmaps of all individual landmarks (see Fig. 1). The input image is resampled to a uniform voxel spacing of 8 mm and centered at the network input. Since the network input resolution is $[64 \times 64 \times 128]$, every volume of the dataset fits into the network.

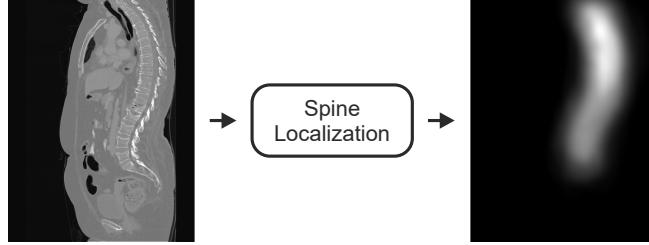


Fig. 1: Input volume and spine heatmap prediction of the spine localization network.

2.2 Vertebrae Localization

To localizes centers of the vertebral bodies, we use the SpatialConfiguration-Net proposed in [2]. The network effectively combines the *local appearance* of landmarks with their *spatial configuration*. The *local appearance* part of the network is based on the U-Net, while the *spatial configuration* part consists of four convolutions with $[7 \times 7 \times 7]$ kernels in a row and is processed in 1/4 of the resolution of the *local appearance* part. For more details of the network architecture and loss function, we refer the reader to [2].

A schematic representation of how the input volumes are processed to predict the final heatmaps is shown in Fig. 2. Every input volume is resampled to have a uniform voxel spacing of 2 mm, while the network is set up for inputs of size $[96 \times 96 \times 128]$. With these volume size and spacing, many images of the dataset do not fit into the network and cannot be processed at once. To narrow the region of interest in the vertebral localization step, we used the predicted of the spine localization network, see Sec. 2.1. Furthermore, as some volumes have a larger extent in the z -axis (i.e., the axis perpendicular to the axial plane) that would not fit into the network, we process such volumes as follows: During training, we



Fig. 2: Input volume and individual heatmap predictions of the vertebrae localization network. The yellow rectangle indicates that not the whole input volume is processed at once, but overlapping cropped sub-volumes. For each possible landmark, an separate heatmap volume is predicted, which is visualized with different colors.

crop a subvolume at a random position at the z -axis. During inference, we split the volumes at the z -axis into multiple subvolumes that overlap for 96 pixels, and process them one after another. Then, we merge the network predictions of the overlapping subvolumes by taking the maximum response over all predictions, for more details please check [2].

We detect the final landmark positions as follows: For each predicted heatmap volume, we detect multiple local heatmap maxima that are above a certain threshold. Then, we determine the first and last vertebrae that are visible on the volume by taking the heatmap with the largest value that is closest to the volume top or bottom, respectively. We identify the final predicted landmark sequence by taking the sequence that does not violate following conditions: consecutive vertebrae may not be closer than 12.5 mm and farther away than 50 mm, as well as a following landmark may not be above a previous one.

2.3 Vertebrae Segmentation



Fig. 3: Input volume and segmented vertebrae of the spine segmentation network. The yellow rectangle shows the cropped region around a single vertebra and indicates that each localized vertebrae is processed individually. Each individually detected vertebra is then merged back to the final multi-label segmentation.

For creating the final vertebrae segmentation, we use a U-Net [3] to segment each localized vertebra (see Fig. 3). The U-Net is set up with a sigmoid cross-entropy loss for binary segmentation to separate individual vertebrae from the background. Since each vertebra is segmented independently, the network needs to know, which vertebra it should segment. Thus, from the whole spine image we crop the region around the localized centroid (see Sec. 2.2), such that the vertebra is in the center of the image. Furthermore, in the same way as the vertebral image, we also cropped a heatmap image of vertebral centroid from the heatmap prediction of the vertebral localization network. Both cropped vertebral image and heatmap image of vertebral centroid are used as an input for the segmentation network. Both input volumes are resampled to have a uniform voxel spacing of 1 mm, while the network is set up for inputs of size $[128 \times 128 \times 96]$.

To create the final multi-label segmentation result, the individual predictions of the cropped inputs are resampled back to the original input resolution and translated back to the original position.

3 Implementation Details

Training and testing of the network was done in Tensorflow¹, while we perform on-the-fly data augmentation using SimpleITK². As data augmentations we use intensity shift and scale, as well as spatial translation, scaling, rotation and elastic deformation. We evaluate training and network hyperparameters with a three-fold cross validation. The results submitted to the challenge were generated with networks that were trained with all 80 annotated training volumes from the VerSe 2019 challenge. The training and testing scripts will be made publicly available³.

4 Conclusion

In this technical report we have proposed a three step fully automatic approach for vertebrae localization and segmentation. The predicted localizations and segmentations submitted to the VerSe 2019 await comparison to other participating methods.

References

1. Glocker, B., Feulner, J., Criminisi, A., Haynor, D.R., Konukoglu, E.: Automatic localization and identification of vertebrae in arbitrary field-of-view CT scans. In: Proc. Med. Image Comput. Comput. Interv. vol. 15, pp. 590–8 (2012)
2. Payer, C., Štern, D., Bischof, H., Urschler, M.: Integrating Spatial Configuration into Heatmap Regression Based CNNs for Landmark Localization. Med. Image Anal. 54, 207–219 (may 2019)
3. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Proc. Med. Image Comput. Comput. Interv., pp. 234–241. Springer (2015)
4. Sekuboyina, A., Rempfler, M., Kukačka, J., Tetteh, G., Valentinitisch, A., Kirschke, J.S., Menze, B.H.: Btrfly Net: Vertebrae Labelling with Energy-Based Adversarial Learning of Local Spine Prior. Proc. Med. Image Comput. Comput. Interv. pp. 649–657 (2018)

¹ <https://www.tensorflow.org/>

² <http://www.simpleitk.org/>

³ <https://github.com/christianpayer/MedicalDataAugmentationTool/>

Report: Verse 2019 Grand Challenge

Christoph Angerman

Department of Mathematics, University of Innsbruck, Technikerstraße 13, A-6020
Innsbruck
christoph.angermann@uibk.ac.at
<http://applied-math.uibk.ac.at>

Abstract. This short report faces solution methodologies for the Verse 2019 Grand Challenge, which is held in conjunction with MICCAI 2019 in Shenzhen (China). The challenge consists of two tasks. Given a spine CT scan, the first task is to find techniques for labelling all the vertebrae within the field of view (landmark detection task). In the second task, the goal is to find a fully-automated methodology for volumetric vertebrae segmentation with multiple output channels (3D multilabel segmentation). For this task, we use deep learning and convolutional networks [1–3, 8]. Given the resulting 3D segmentation masks, we are able to handle the first task, computing weighted centroids of each vertebra individually.

1 Introduction

Deep convolution neural networks have become a powerful method for image recognition [4, 8]. In the last few years they also exceeded the state-of-the-art in providing segmentation masks for 2D images. Long et al. [5] proposed the idea to transform VGG-nets [8] to deep convolution filters for obtaining semantic segmentations of 2D data. Based on these deep convolution filters, Ronneberger et al. [7] introduced a novel network architecture, the so-called U-net. With this architecture they redefined the state-of-the-art in image annotation till today. The U-net provides a powerful 2D segmentation tool for biomedical applications, since it has been demonstrated to learn highly accurate segmentation masks from only very few training samples.

Among others, the fully automated generation of volumetric segmentation masks is becoming more and more important for biomedical applications. This task is still challenging. One idea is to extend the U-net structure to volumetric data by using 3D convolutions, as has been proposed in [2, 3, 6]. Significant drawbacks of 3D convolution models are the huge memory requirements and the resulting restrictions to the model’s complexity. Deep learning segmentation methods are therefore often applied to 2D slice images [2]. However, these slice images do not contain information of the full 3D data, which makes the segmentation task much more challenging.

To address the drawbacks of existing approaches, in [1] the authors introduced a network structure which is able to generate accurate 3D segmentation

masks of very large volumes. The main idea is to include projection layers from different directions which transform the data to 2D images containing information of the full 3D data. Then a 2D U-net is applied to these projection images before a learnable reconstruction algorithm is used to lift them again to volumetric data.

The combination of the mentioned slice-by-slice approach with the projections-based methodology in [1] is our solution for the task of providing a fully-automated technique for volumetric spine segmentation (task 2). Note that for our methodology we split task 2 into two sub-tasks:

- The goal of the first sub-task is to find a method for 3D spine segmentation with only one output channel, which gives the probability that a voxel denotes a vertebra (Fig. 1a).
- Considering the second sub-task, we search for a technique to assign the labels C1-L6 to the segmented vertebrae out of sub-task 1 (Fig. 1b).

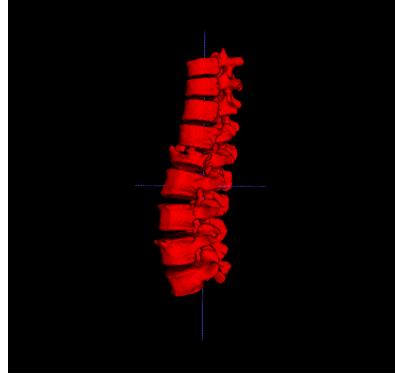


Fig. 1a: Segmentation mask with only one output channel.

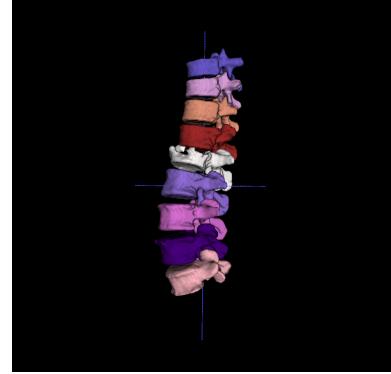


Fig. 1b: Segmentation mask with an output channel for each vertebra.

For the task of vertebrae labelling (task 1), we use the 3D segmentations of task 2 and compute weighted centroids for each label (i.e. vertebra) individually.

2 Methodology

2.1 Vertebrae Segmentation

In the first step, we read in a 3D CT spine scan, save its spacing, orientation and origin, and convert the scan to a three-dimensional array. Note, that different input scans have different shapes. Therefore, we zoom the array such that the longest axis is now of size 224 and pad the input array with zeros to obtain a

cubical array in $\mathbb{R}^{224 \times 224 \times 224}$. So we can ensure equal input sizes for our convolutional segmentation models.



Fig. 2: MIP images of a 3D spine scan with directions $\{k \times 30 \mid k = 0, \dots, 5\}$.

We split the task of multi-label vertebrae segmentation into **two sub-tasks**:

1. In the first step, the goal is to map each voxel of the input CT scan to the probability of being a foreground voxel (i.e. denoting a vertebra). The result should be a one-channel segmentation mask as in Fig 1a. In this step, we do not take care of the different labels of the vertebrae.

For this aim, we make use of the 2.5D U-net structure proposed in [1]. So the model takes as input the 3D array and generates 2D projection images, which contain information of the full 3D data. For the targeted application of spine segmentation, we make use of the *Maximum Intensity Projection* (MIP) (Fig 2). We propagate these 2D projection images through a 2D U-net [7] and lift them again to volumetric data using a trainable reconstruction algorithm [1, Equation (III.1)].

Since the choice of the MIP as projection technique only generates satisfying results for convex objects , we suggest to combine the resulting segmentation of the 2.5D U-net with the segmentation generated by a slice-by-slice approach. Therefore, we again train a 2D U-net with nearly 2.2×10^6 parameters, which outputs for each slice of the 3D input the probabilities, that the pixels of the slice denote vertebrae. Stacking these segmented slice images to a 3D array again also results into a volumetric segmentation of the 3D array.

Combination of the 2.5D U-net and the slice-by-slice 2D U-net approach delivers the desired 3D segmentation masks (Fig. 1a).

2. After we derived a fully-automated method for indicating which of the input voxels denote vertebrae, we look for a methodology to label these detected vertebrae. Therefore, we again train a 2D U-net [7] with nearly 138×10^6 adjustable parameters and 26 output channels. This model outputs for each pixel in a 2D MIP image a vector of probabilities, which denotes to which label C1-L6 the detected vertebra belongs. We apply the trained labeling model to MIP images (Fig. 2) with angles in $\{0^\circ, 10^\circ, 80^\circ, 90^\circ, 100^\circ, 170^\circ\}$. So all in all, we obtain 6 labeled MIP segmentations. The labeling in 3D

is then done by backprojecting the MIP segmentations. To be precise, we multiply them along the first axis with the 3D segmentation of the first sub-task, rotated with the corresponding angle. This results into the desired multilabel segmentation in Fig. 1b.

2.2 Vertebra Labelling

For this task, we make use of the 3D multilabel segmentations we generated in Section 2.1. We do not propose any trainable models here. We just look at each labeled vertebra individually, weight the edges of the vertebra and compute the centroid. The weights of the edges are set empirically and are for all vertebrae the same. So we obtain for each vertebra, which has been detected in the segmentation task, the desired centroid coordinates and of course the corresponding label, since they already have been labeled in the segmentation task.

3 Conclusion and Future Work

To sum up, we proposed fully-automated methodologies to handle both tasks, the volumetric multilabel segmentation of spine CT scans and the labelling of vertebrae. Unfortunately, we are not satisfied with the performance of our multilabel segmentation methodology so far. Therefore, in future work we will try to increase performance of our segmentation models, especially for the 2D labeling U-net in Section 2.1 a performance boost is necessary. Furthermore, we think that we lose too much accuracy if we zoom all images to cubic arrays of size 224. We will work on more sophisticated techniques to handle the different input shapes of the spine CT scans

References

1. Angermann, C., Haltmeier, M., Steiger, R., Pereverzyev Jr, S., Gizewski, E.: Projection-based 2.5 d u-net architecture for fast volumetric segmentation. arXiv preprint arXiv:1902.00347 (2019)
2. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
3. Erden, B., Gamboa, N., Wood, S.: 3D convolutional neural network for brain tumor segmentation (2018)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
6. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 565–571. IEEE (2016)

7. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

Large Scale Vertebrae Segmentation Using nnU-Net

Yujin Hu¹, Baiying Lei¹, and Tianfu Wang¹

The National-Regional Key Technology Engineering Laboratory for Medical Ultrasound, Guangdong Key Laboratory for Biomedical Measurements and Ultrasound Imaging, School of Biomedical Engineering, Health Science Center, Shenzhen University, Shenzhen, China tfwang@mail.szu.edu.cn
<http://bme.szu.edu.cn/>

Abstract. Semantic segmentation is an important and popular subfield in medical image analysis with a vast number of new methods being proposed each year. However, many proposed methods fail to generalize beyond the experiments they were demonstrated on. Recently, Fabian proposed nnU-Net, a framework that automatically adapts itself to any given new dataset. nnU-Net attempt to automate necessary adaptations, such as preprocessing, the exact path size, batch size, and the inference settings based on the properties of a given dataset.

Keywords: nnU-Net · semantic segmentation · Generalization.

1 Method

We do not propose new method, this is a simple application of nnU-Net [1]. We briefly describe nnU-Net, readers are suppose to read the original paper for detail. The official is available at GitHub¹, and our fork with proper modification to run without GPU as required for this challenge will be public available², too.

1.1 Preprocessing

For CT images, all foreground voxels in the training set are collected and an automated level-window-like clipping of intensity values is performed based on the 5 and 99.5 percentile of these values. The data is then normalized with the global foreground mean and standard deviation. The described scheme is independently applied to each case. nnU-Net also collects all spacings within the training data and for each axis chooses the median as the target spacing. All training cases are then resampled with third order spline interpolation. However, the VerSe2019 dataset seems has same or similar spacing in all cases, this resample scheme might have little impact on the final result.

¹ <https://github.com/MIC-DKFZ/nnUNet>

² <https://github.com/hubutui/nnUNet>

1.2 Training Procedure

Network Architecture: Three U-Net models are configured, designed and trained independently: a 2D U-Net, a 3D U-Net and a cascade of two 3D U-Net where the first generates a initial low resolution segmentation which is subsequently refined by the second model. Note that nnU-Net uses padded convolutions to achieve identical output and input shapes, which is common used. And nnU-Net uses instance normalization and Leaky ReLU instead of batch normalization and ReLU, respectively. *Network Hyperparameters:* nnU-Net automatically sets the batch size, patch size and number of pooling operations for each axis while keeping the GPU ram consumption with a certain budget (12 GB NVIDIA TitanXp GPU). In addition, large patch size are favored over large batch size (with a minimum of 2) to maximize the amount of spatial context that can be captured. Pooling along each axis is done until further pooling would reduce the spatial size of this axis below 4 voxels. All U-Net architectures use 30 convolutional filters in the first layer and double this number with each pooling operation. If the selected patch size covers less than 25% of the voxels in case, the 3D U-Net cascade is additionally configured and trained on a downsampled version of training data. *Network Training:* All models are trained in a five-fold cross-validation. The sum of the cross-entropy loss and the dice loss are used as loss function. Adam was used as optimizer for stochastic gradient decent with an initial rate of 3×10^{-4} and l_2 weight decay of 3×10^{-5} . The learning rate is dropped by a factor of 0.2 whenever the exponential moving average of the training loss does not improve within the last 30 epochs. Training is stopped when the learning rate drops below 10^{-6} or 1000 epochs are exceeded. For data augmentation, nnU-Net uses elastic deformations, random scaling and random rotations as well as gamma augmentation.

1.3 Inference

All cases are predicted using a sliding windows approach with half the patch size overlap between predictions. And test time data augmentation is applied by mirroring along all axes. Original nnU-Net ensembles combinations of two U-Net configurations (2D, 3D, and cascade 3D). However, we could not finish the whole training before deadline due to limit computation resources. Our submission will ensemble only 3D U-Net with full resolution input image, 3D U-Net with low resolution input image (part of the cascade 3D), and 2D U-Net. We would like to ensemble all models in test phase 2.

References

1. Isensee, F., Petersen, J., Kohl, S.A., Jäger, P.F., Maier-Hein, K.H.: nnU-Net: Breaking the spell on successful medical image segmentation. arXiv preprint arXiv:1904.08128 (2019)

An Automatic Multi-stage System for Vertebra Segmentation and Labelling

Maodong Chen, Xi Cheng, and Dalong Cheng^{*}

iFLYTEK Research South China, Computer Vision Group
11F, Tower 1, Heung Kong International Innovation Center, Nansha District,
Guangzhou, China
dlcheng2@iflytek.com

Abstract. We design and implement a multi-stage system to segment and label vertebrae. Two 3D U-Nets are used in a sequence for the segmentation, which generate ID related masks referring to large patches of the image at first and then refine them according to the inferred masks. It works in the spirit of gathering broad information for the vertebrae identification and refining each instance mask locally. We then match the refined instance masks to the masks with labels by considering the continuity of the spines and the overall overlap between the two stages. The center of the spine is regressed by another network using the RCNN (region CNN) strategy, where the ROI (region of interest) is cropped according to the segmentation mask. We find this multi-stage system easy to implement and tune, which gives competitive results in the first test phase of the VerSe2019 challenge.

1 Introduction

Vertebral segmentation and localization[1, 2] provide direct understanding of the anatomical structures and have significant implementations in helping to detect kyphosis or scoliosis, vertebral fractures and in analyzing the tasks of surgical plans.

The Large Scale Vertebrae Segmentation Challenge (VerSe2019) in conjunction with MICCAI 2019, has established a reasonably rich public dataset in spine CT scans and aims to stimulate teams in developing methods for both labelling and segmentation tasks. Here we present our approach to this challenge.

2 Methodology

2.1 Data Preprocessing

VerSe2019 has 80 scans for training. The in-plane resolution is 1.0 mm, while the slice spacing varies from 1.0 to 3.0 mm. We perform three types of preprocessing

^{*} Maodong Chen and Xi Cheng made equal contribution to the project. Dalong Cheng conducted and supervised the research. Corresponding author: Dalong Cheng

on the dataset: (1) the image axes are rearranged to the RAS+ orientation (left to Right, posterior to Anterior, inferior to Superior); (2) the voxel intensity is normalized by subtracting the mean and dividing by the standard deviation of the complete dataset; (3) the spatial resolution is resampled to 1 mm by the third-order spline interpolation, which makes all cases isotropic.

2.2 The Segmentation

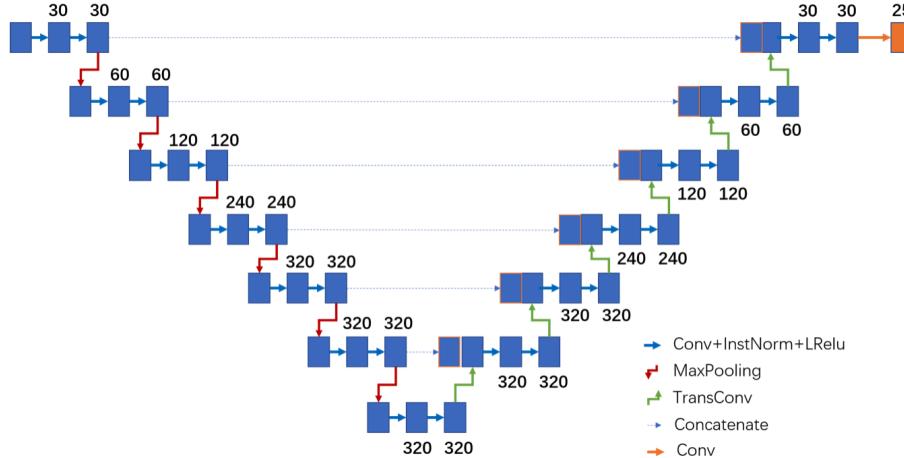


Fig. 1. The 3D U-Net structure of segmentation stage-1. The input size is 224*160*128 and the output has 25 channels corresponding to the 24 spine labels and 1 background label. The second stage has a similar structure, except that it is shallower and the output is binary in channels.

We adopt a two-stage strategy to predict the masks. The first stage is a 3D U-Net as shown in Fig. 1. We feed the network with patches of size 224*160*128 extracted randomly from each scan. The stage-1 network is demanded to predict 25 labels (we ignore the L6 label as it is rare.) according to the ground truth of each patch. We argue that using these patches as input other than the whole scans brings more variety in the training samples and makes it more easy to capture the essential feature of each spine class and helps the identification.

The segmentation stage-1 generally performs well on instances close to C1 or L5 and keeps the foreground mask as well as the counting number correct. However, the predictions of the middle ones are more easily mixed with each other, since they are similar. To resolve this problem, we propose a second stage refinement network, whose structure is very similar to the stage-1 network. We make changes mainly in the training of stage-2. We generate patch samples that cover a spine in the middle and extend 1.5 times more in the slice direction, which generally contains other spines. We pad the patch to 128*128*128 with

zeroes if necessary and force the network to predict only the middle spine with binary labels (foreground or background).

The second stage is straight forward to intergrade with the labeled masks from stage-1. Due to the training policy of stage-2, it intends to predict a instance mask seldomly mixing with neighbors. We first merge the stage-1 masks to a binay mask indicating the foreground. Then each stage-1 mask is used to generate patch for stage-2. The prediction from stage-2 is believed to be more accurate in instance level and filled to the binary foreground only if the new mask overlaps little with exsiting ones. If all stage-1 masks have been reviewed, while the foreground is still not filled sufficiently, new patches will be selected from the not-filled regions for stage-2 untill convergence.

Because the well segmented instances in stage-1 and stage-2 shall mostly overlap, it is operable to assign shared labels from stage-1 to stage-2 by comparing the dice of the pairs. With the constraint on the label continuity of neighboring spines, this process can be easily performed. Here we present a matching mechanism as follows:

Algorithm 1: update label for stage-2 vertebrae set

Input: Stage-2 vertebrae set V_n ($V_n=1,2,\dots,k$) and the stage-1 vertebrae set V_r (size= $m, 1 \leq \max(V_r) \leq 26$)
if stage-1 vertebrae set contain label 22 or 23 and $m \leq 12$ **do**
 for instance $i \in V_n, i=k, k-1$ **do**
 for vertebra $v_j \in V_r, v_j \geq i$ **do**
 Calculating and recording dice index for instance i with vertebra v_j
 Find the Maximum of record dice and the corresponding vertebra v_b
 if maximum of record dice ≥ 0.8 **do**
 if $i=k$ **do**
 update label for stage-2 vertebrae set from v_b-k+1 to v_b
 else
 update label for stage-2 vertebrae set from v_b-k+2 to v_b+1
 break
 else
 for instance $i \in V_n, i=2,3,4$ **do**
 for vertebra $v_j \in V_r, v_j \geq i$ and $v_j \leq 25-k+1$ **do**
 Calculating and recording dice index for instance i with vertebra v_j
 Find the Maximum of record dice and the corresponding vertebra v_b
 if maximum of record dice ≥ 0.8 **do**
 if $i=2$ **do**
 update label for stage-2 vertebrae set from v_b-1 to v_b+k
 else if $i=3$ **do**
 update label for stage-2 vertebrae set from v_b-2 to v_b+k+1
 else
 update label for stage-2 vertebrae set from v_b-3 to v_b+k+2
 break

2.3 The Localization

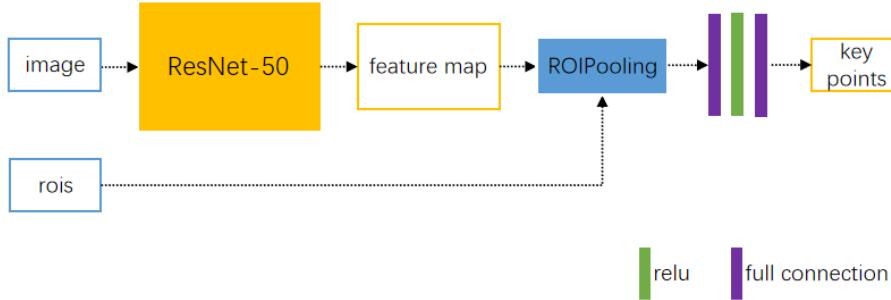


Fig. 2. Scheme of the RCNN localization network.

We use a RCNN[4, 5] like architecture to deal with the center localization problem of VerSe2019. As shown in Fig. 2, a 3D ResNet-50 is used as the backbone. On feature map of stride 4, ROI pooling is performed to extract features to regress the deviation of the spine center to the ROI box center in unit of the box size. The input of network has a size of $160 \times 192 \times 224$. Different from the traditional RCNN with classification scores, we don't need negative samples. In the training phase, boxes are generated from the segmentation ground truth. We tend to give some freedom of these boxes to make more positive samples. Another thing important we notice is that the spine center is defined in the main body. However, the enclosed box from direct segmentation mask has to include other parts such as the spinous process, making the regression difficult. We define a much tight ROI box by examining the largest connected regions in the sagittal direction, which correspond more reasonably to the main body of a spine. Our localization modual needs the segmentation in the inferring, so it can be viewed as stage-3 of the total workflow.

3 Experiments and Results

The U-Nets of stage-1 and stage-2 are realized using the nnUnet package [3]. We train the networks using Adam optimizer, weight decay of 3×10^{-5} and an initial LR of 3×10^{-4} . The LR would drop by a factor of 0.2 if the moving average training loss does not improve over the last 30 epochs. We stop the training if LR is less than 10^{-6} . The summation of CE and dice loss is used. Elastic deformation, random scaling and random rotation are performed as data augmentation in both training and testing phases. The RCNN network is trained with batch size of 4 and initial LR of 10^{-3} . The LR decreases by a factor of 0.1 after the 15th and 20th epoch, where we totally train 25 epochs. For each image, we sample at most 200 ROIs as positive samples. We apply random scaling,

flipping and hard samples mining during the training. Smooth L1 is used as the loss function. No test augmentation is used for regression.

Methods	Val Perf		cv_fold_1		cv_fold_3		cv_fold_4	
	Dice	HD-Dist	Dice	HD-Dist	Dice	HD-Dist	Dice	HD-Dist
Without Refinement model	0.884	20.672	0.831	18.605	0.921	17.945		
With Refinement model	0.953	16.119	0.921	21.623	0.964	4.588		

Table1: Results with and without refinement of stage-2 on VerSe2019 train dataset. **cv_fold_i** ($i=1,2,3,4,5$) refers to the indicated fold of the cross validation. **HD-dist** refers to the Hausdorff distance.

Methods	Val Perf			cv_fold_1			cv_fold_3			cv_fold_4		
	ID-rate	MLD/mm	Recall	Precision	ID-rate	MLD/mm	Recall	Precision	ID-rate	MLD/mm	Recall	Precision
Without Refinement model/Without roi refinement	0.99	7.62	0.98	0.84	0.93	7.59	0.94	0.81	0.96	7.48	0.95	0.80
With Refinement model/With roi refinement	0.99	3.51	0.98	0.95	1.00	3.64	1.00	0.96	1.00	3.65	1.00	0.97

Table 2: Results on the localization metrics without any refinement and with all refinement (stage-2 and tight ROI). The **cv_fold_i** has the same meaning as in Table 1.

Fig. 3 shows the effectiveness of the segmentation stage-2. The refinement corrects the label mixing by producing more accurate instance masks and rearranges the labels regarding to the continuity relation from the matched spines at the bottom. Table 1 compares the segmentation metrics before and after stage-2 on three folds of the cross validation. Refinement improves the dice by a large margin and in general reduces the Hausdorff distance. In the submission phase, we use the model created in first fold as it is smoother and more consistent. Table 2 compares the metrics of localization before and after applying stage-2 and tight ROI. It shows that these two mechanisms together are significantly helpful in reducing the regression error and lowering the false positive rate.

4 Conclusion

We developed a multi-stage system to tackle the vertebra segmentation and localization problems. For the segmentation, we implemented a coarse-to-fine strategy

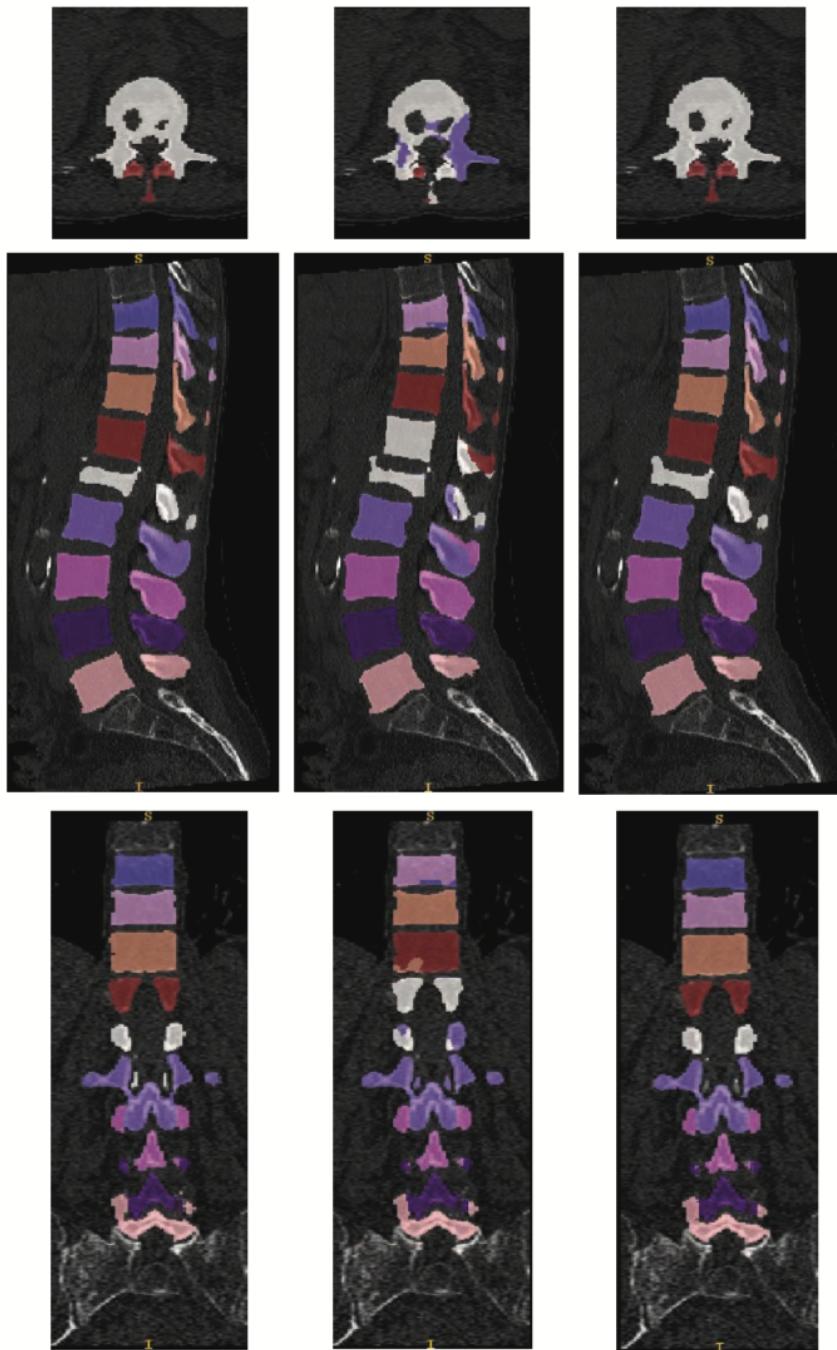


Fig. 3. Illustration of the label mixing in stage-1 (middle column) and the correctness after applying stage-2 (right column). The left column corresponds to the ground truth. The rows represent different views of the scan.

to improve the results, where the continuity relation was embedded explicitly. For the localization task, we trained a RCNN using segmentation generated boxes. Overall, we showed that this multi-stage system could be transparent to train and accurate to use.

References

1. Glocker, B., et al., Automatic localization and identification of vertebrae in arbitrary field-of-view ct scans. In: MICCAI. (2012)
2. Sekuboyina A, et al., Btrfly Net: Vertebrae Labelling with Energy- based Adversarial Learning of Local Spine Prior. In: MICCAI. (2018)
3. Isensee, Fabian, et al., nnU-Net: Breaking the Spell on Successful Medical Image Segmentation, arXiv:1904.08128 (2019)
4. Ross Girshick, et al., Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR14, 580-587 (2014)
5. Ross Girshick, Fast R-CNN, arXiv:1504.08083 (2015)

Verse 2019 Method Report: Team INIT

Xin Wang and Qingyue Wei

Department of Electronic Engineering, Fudan University
Department of Radiology, University of North Carolina at Chapel Hill

Abstract. This report describes the methods we use to solve the two problems regarding vertebra labelling and segmentation in Verse 2019 challenge. In order to label vertebrae, we firstly implement a Single Shot MultiBox Detector (SSD) to crop the original 3D images into vertebra-centered versions. Then we obtain the 2D sagittal and coronal projections as the inputs of the Btrfly-net, from which we reconstruct the labelling results. As for Task 2, we crop the original images into regions that are around the labelling locations from Task 1 and a 3D U-net with residual blocks is used to segment vertebrae, the local regions as inputs. In this situation, U-Net does not need to discriminate different bones, and hence the performance is boosted.

Keywords: Btrfly-net, 3D U-net, Residual Learning, SSD

1 Task 1: Labelling

1.1 Basic Network: Btrfly-net

In light of good performance obtained by Btrfly-net, it is advisable to reproduce the network and then look for further helpful algorithms. Hence, we firstly build the Btrfly-net structure according to the Btrfly paper, which takes the sagittal and coronal projections of 3D CT scans as input and produces the corresponding heat maps that can be used to reconstruct the positions of labels. We have yet to implement GAN described in the same paper due to limited time, but further improvement is ongoing.

1.2 Preprocessing

We have finished several experiments in order to find the factors that the Btrfly-net is sensitive to, and then try to optimize those factors for better performance. There are several points according to our pre-experiments and the Btrfly paper. (1) It will be better if the input images of Btrfly-net are vertebra-centered. (2) The input images need to have an isotropic resolution in all the dimensions so that the network can detect the bones more easily.

In addition, batches must be used in order to accelerate training speed and reduce error rate of batch normalization (according to Kaiming He's paper, group normalization is better in the situation of small batch size compared to batch normalization), which means the sizes of input images must be identical.

All the points mentioned above are considered in preprocessing and we will describe how to achieve these effects in the following sub-subsections.

Cropping As for vertebra-centered images, a Single Shot MultiBox Detector (SSD) is implemented to localize the vertebrae in the sagittal and coronal projections and the results are used to crop the original 3D images.

Resampling The cropping results are resampled to an isotropic resolution of 1mm in all the dimensions. We set the value to 1mm for that the original images are mostly 1mm-resolution, which means we can reserve most of information during resampling.

Padding The sizes of those resampling results are still various. In that case we cannot use batches to speed up training process and the concat inside the Btrfly-net will fail due to different sizes. Hence, we add zero paddings to the resampled images so that the inputs of the Net are 610×610 images.

1.3 Calibration

The Btrfly paper proposed a kind of method to reconstruct vertebra positions from the outputs of the Net, which is to calculate argmax of the tensor product of the two output heat maps. This method, however, sometime causes error due to the quality of inputs, especially when the coronal projections themselves are so opaque that even a human may have no ability to localize bones on them.

Three numbers are needed to form 3D coordinates, two of which are related to x-y plane and the last one are related to z-axis. Let us say the z-axis is the direction along the spine. Then, we found the predicted positions are accurate with respect to x-y plane but usually inaccurate in the third (z) direction.

In order to revise this, considering the heat map of the coronal projection and that of the sagittal projection are regarding x-z plane and y-z plane, respectively, we propose another method to obtain the third coordinate: instead of getting it from the tensor product, we firstly find the argmaxes from the two 2D heat map (let us say they are (x, z_s) in the sagittal heat map and (y, z_c) in the coronal heat map), then calculate the final z coordinate from the weighted average of z_s and z_c , the max values of the two heat maps as the two weights. By implementing this algorithm, we obtained an increase of 4% on identification rate.

Moreover, we also noticed that some of the positions are missed in the prediction results, while they can be easily predicted according to the positions of those vertebrae around them, and normally the labels in one single subjects are consecutive. Therefore we add interpolation to predict the locations of missed vertebrae.

2 Task 2: Segmentation

2.1 Method

Nowadays U-net has been widely adapted in the task of biomedical image segmentation. Since the spine CT scan data is in 3D volume, a 3D U-net based

architecture has been used in our task. In order to prevent the vanishing gradient problem. We also combine 3D U-net with residential blocks.

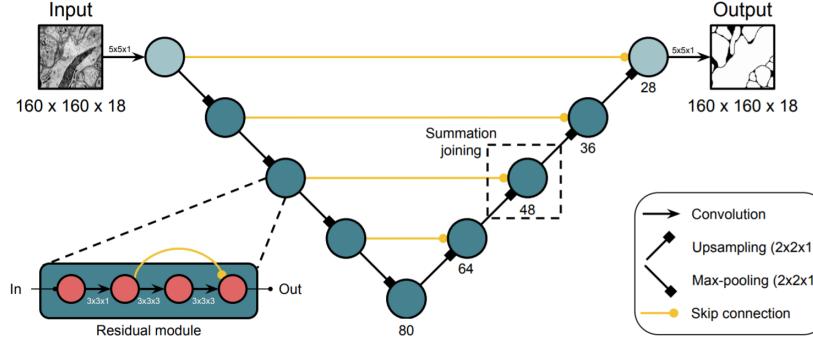


Fig. 1. The structure of 3D U-net with residual blocks.

For the sake of separating vertebrae from the background and annotating it with labels simultaneously, we have tried 2 different strategies: (1) use the network to do both segmentation and multi-labelling work. (2) since the centroid of each spine has been labelled, the network only needs to identify each vertebra when given its corresponding centroid position.

However, the problem is that the available data is unbalanced and scarce as well. It might be tough for a single network to do the multi-label segmentation, so we finally choose strategy 2 to do the segmentation.

Structure of 3D-U-net with Residential Blocks The network is based on 3D U-net and also combined with residential blocks. The architecture consists of 5 encoder blocks and 4 decoder blocks. Thus, there exists 5 levels. A summation joining has been done in each level as a skip connection. There is a skip connection in each encoder and decoder block making it a residual module.

Loss Function We choose a dice loss as our loss function. The loss is computed in each channel. The total loss is expressed as:

$$\text{Dice Loss} = \frac{2|S_{pred} \cap S_{true}|}{|S_{pred}| + |S_{true}|} \quad (1)$$

where S_{pred} and S_{true} are the output of U-net and the ground-truth segmentation, respectively, and $|S|$ represents the number of elements in the set S . Since the multi-label segmentation task is simplified as a binary segmentation, the pixel value in ground-truth will be equal to 1 or 0 only. The dice losses are calculated in each channel and the final loss is their mean value.

2.2 Experiments

Dataset The total number of the CT scans is 78. 20% of the data is used as validation set and the rest is training set. To identify a single vertebra, the input is a $96 \times 96 \times 96$ 3D volume from CT scans taking the centroid position as its center.

Data Preparation It could be seen that the value range of original CT scans is at big scale. Thus, before putting the data into the network, we normalize the data first. Besides, to make the edge of each vertebrae more distinct, we also adapt sharpening in the preprocessing.

Result Construction Suppose we have a $w \times h \times d$ spine CT scan. To begin with, we create two null matrixes in size $w \times h \times d$: one is for the final segmentation result and the other is for recording the scores. According to the result of labelling detection, we can get all the labels and corresponding coordinate. Segmentation of each label has been done separately. For each label, the output has two channels, thus every pixel in the input has two different logits. If logits in channel 0 is larger than channel 1, it turns out the pixel belongs to the 'background' (including the real background and pixels from other labels) and vice versa. Once the pixel is predicted to be part of label i, the pixel in the same position in the final result matrix will equal to i. However, if there is a conflict that this pixel has already been assigned as j, compare their corresponding logits. If logit of i is higher, then replace j with i in the result matrix and the same in the score recording matrix. Repeat until all the label is evaluated by the network and the final result matrix will be the result of the multi-label segmentation.

References

1. Sekuboyina A. et al. (2018) Btrfly Net: Vertebrae Labelling with Energy-Based Adversarial Learning of Local Spine Prior. Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. Springer, Cham.
2. Roy, A.G., et al.: Error corrective boosting for learning fully convolutional networks with limited data. In: MICCAI. pp. 231–239. Springer (2017)
3. K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung. Superhuman accuracy on the snemi3d connectomics challenge. arXiv preprint arXiv:1706.00120, 2017.
4. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In ECCV, 2016.

Verse 2019 Challenge

Alexandre Kirszenberg, Nicolas Boutry, Guillaume Tochon, and Élodie Puybareau

EPITA Research and Development Laboratory (LRDE)
Le Kremlin-Bicêtre, France
thierry.geraud@lrde.epita.fr

Abstract. We present a method for automatic vertebrae segmentation and labelling from MDCT scans. This method was developed as part of the Verse 2019 challenge.

Keywords: image segmentation · vertebrae detection · fully convolutional network

1 Introduction

In this report, we propose a method for segmenting and classifying vertebrae from MDCT scans. This method was developed as part of The Large Scale Vertebrae Segmentation Challenge (VerSe2019) .

1.1 The data

The data comprises of 160 MDCT scans around the region of the vertebral column. These scans are separated into three groups:

- A public training set of 80 scans, accompanied with the corresponding labelled ground truth segmentation;
- A public test set of 40 scans;
- A private test set of 40 scans.

The scans are provided in the NIfTI format. The orientation, scale and rotation of the images can vary.

The segmentations are also provided in the NIfTI format, where values can range between 0 and 25. 0 corresponds to background, while values within the range 1-25 correspond to vertebrae labels.

The data is the property of Department of Neuroradiology, School of Medicine, Technical University Munich, 81675 Munich, Germany, and is released under the CC BY-SA 2.0 license.

1.2 The tasks

The challenge is divided into two sub-tasks:

- The first task consists in predicting the centroid of each vertebrae present within the scan, associating it with its corresponding vertebra label;
- The second task consists in predicting the corresponding labelled segmentation.

2 Our method

During this challenge, we focused solely on the second task, since the first task is a sub-problem of the second, and can be inferred directly from the latter's results.

2.1 Vertebrae segmentation

In order to generate a segmentation of the vertebrae, we trained three different U-Net[3] models, using the pseudo-3d segmentation technique from [4]. Each scan was sliced along the three axes to form sets of 3-voxels-wide slices, which provided the input to our models.

Since the dimensions of the input of our models are fixed but the scans' are not, we picked a minimal volume that fit every scan in both the training and the public data set. The dimensions of this volume are (80, 128, 128). The dimension of each axis is a multiple of 16, which is necessary in order for the downsampling/upsampling pipeline of the U-Net model to work properly, where each layer is progressively downsampled, then upsampled back by a factor of 2, a total of 4 times, which gives $2^4 = 16$.

From this minimal volume, we can determine the dimensions of the slices we feed to our models. For each axis, respectively:

1. Sagittal slice: (3, 128, 128);
2. Coronal slice: (80, 3, 128);
3. Axial slice: (80, 128, 3).

In order to train our model, we divided the provided training set into two smaller sets:

- A training set comprising of $2/3$ of the scans (53);
- A validation set comprising of $1/3$ of the scans (27).

After training for ~ 100 epochs on the training set, each of our three models reached a Dice coefficient of 0.91 on the validation set.

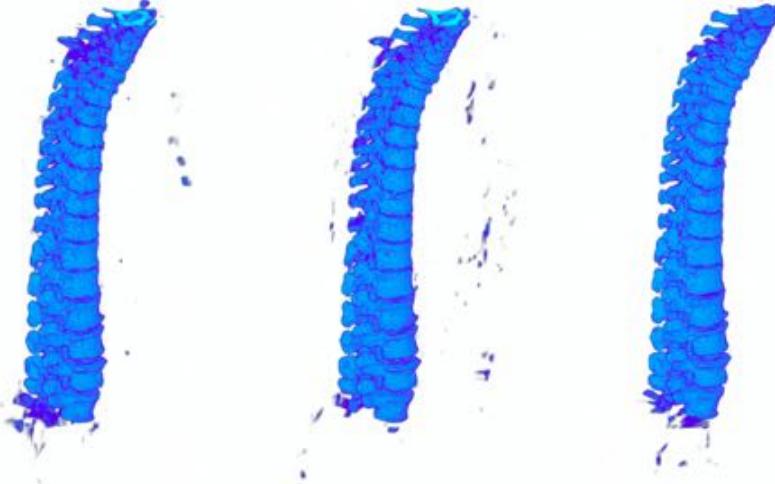


Fig. 1. Predictions on verse095 by each of our three models, which were fed slices from different axes

In order to predict the segmentation of a given scan, we ensure that the whole volume is covered by our slices. Since the dimensions of our volumes are not an exact multiple of our minimal volume, there is some redundancy in the prediction of the voxels that are positioned at the intersection of two overlapping slices.

These predictions are averaged to output a final volume where the value of each voxel, between 0 and 1, represents the probability p of there being a vertebra at its location. Figure 1 shows the prediction output of each of our three models.

We then binarise the predictions with a threshold of $p > 0.5$, and proceed to select only those voxels that were predicted by at least 2 of the 3 models.

Finally, the segmentation is ran through a post-processing step where we filter out smaller objects and fill holes through the segmentation. Empirically, we found that this step improved the final segmentation ever so slightly. Figure 2 showcases the result of this step.

2.2 Vertebrae classification

Once we've segmented all vertebrae, we move on to classify them.

The first step of our algorithm is to detect the spinal cord. To do so, we apply a gaussian blur to the segmentation, then skeletonize it using the `skimage.morphology.skeletonize_3d` method. This provides us with a rough skeleton of our vertebral column, which we further clean up by selecting only the longest

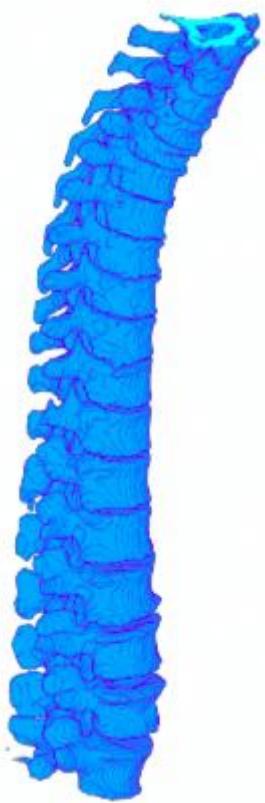


Fig. 2. Prediction on verse095 after the post-processing step

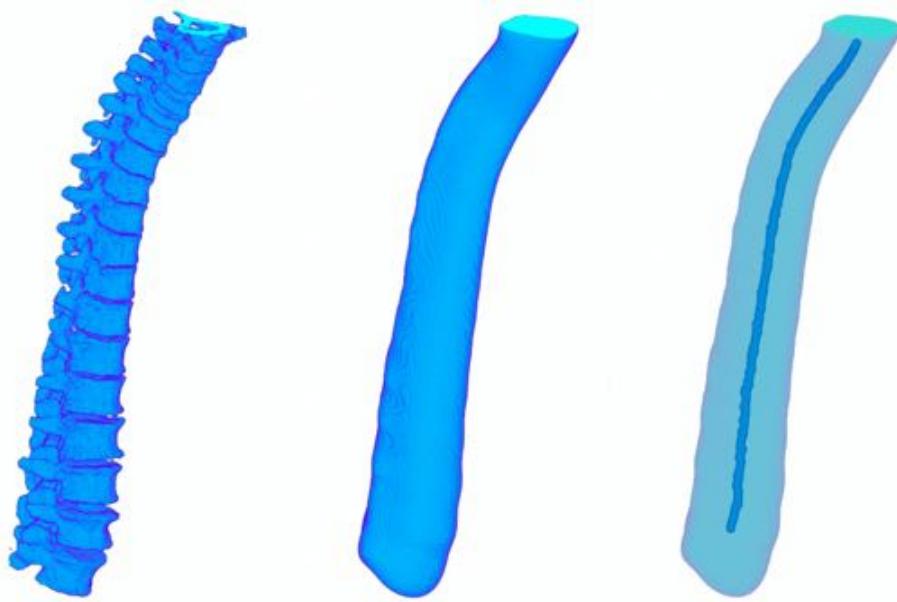


Fig. 3. Detecting the spinal cord skeleton

connected line of voxels between two endpoints with Dijkstra's shortest path algorithm. The result of this process is shown in Figure ??.

While analysing the training data, we found that the centroids of most vertebrae would be found lying exactly on that skeleton. Figure 4 showcases this result for the scan `verse096`.

Our algorithm then transforms this skeleton into a B-spline using SciPy's `interpolate` module, which furthers smoothes and simplifies it, and then discretises it into a series of 1-millimeter-distant points.

On each of these points, we align templates of each vertebrae and multiply them with our segmentation, where the background has been set to -1 to penalize negative space. We select the 5 best (vertebrae, point) candidates, and try to match the previous and next vertebrae before and after the point on the curve, respectively. Once we can't match any other vertebrae, we sum the scores of each vertebrae from each of the 5 vertebral columns, and select the column with the highest score. Then, for each voxel of the column, we select the label of the vertebrae template with the highest score.

The vertebrae templates have been generated from the training data, where we selected each vertebrae of a single label, centered them on their centroid, applied different rotations, and averaged them to produce a kind of probability

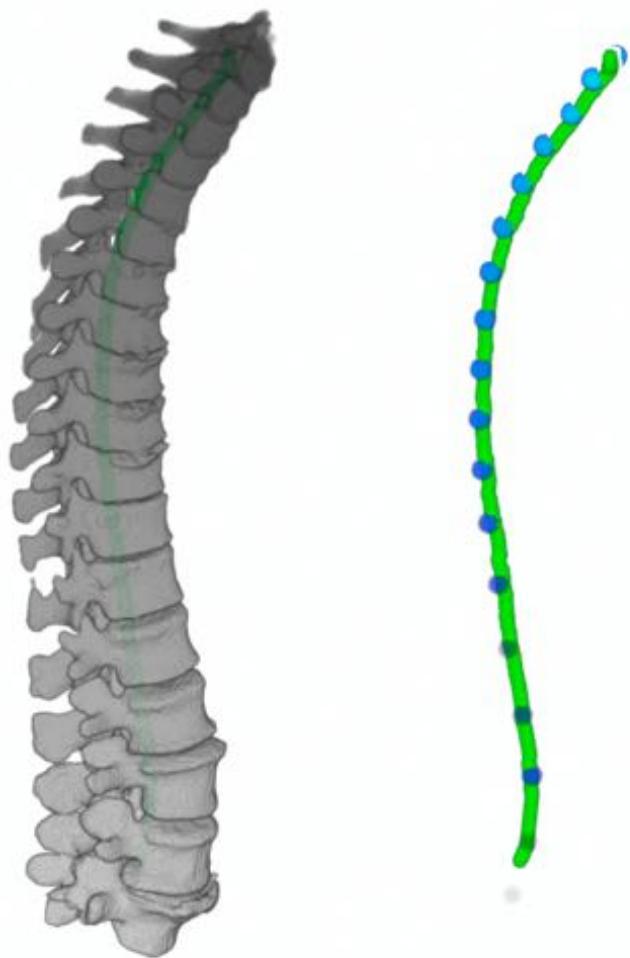


Fig. 4. Centroids are aligned on the detected spinal cord



Fig. 5. The template of vertebra L4

template for this specific vertebra. Figure 5 showcases the result of this process for vertebra L4.

While this classification method produces good results on some scans, it also fails to properly classify most scans, and is prone to confusing close vertebrae.

2.3 Conclusion

While we were able to successfully extract vertebrae from the background using a fully-convolutional neural network architecture based on U-Net, our classification step is not robust enough to be reliable.

We believe an approach centered on separating vertebrae, *then* classifying them—instead of separating and classifying them in the same step—would yield better results. And, more importantly, we would be able to iterate on and improve both algorithms separately.

References

1. Glocker, B., et al.: Automatic localization and identification of vertebrae in arbitrary field-of-view ct scans. In: MICCAI. (2012). https://doi.org/10.1007/978-3-642-33454-2_3
2. Sekuboyina A, et al.: Btrfly Net: Vertebrae Labelling with Energy- based Adversarial Learning of Local Spine Prior. In: MICCAI. (2018) https://doi.org/10.1007/978-3-030-00937-3_74
3. Olaf Ronneberger and Philipp Fischer and Thomas Brox: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI. (2015) https://doi.org/10.1007/978-3-319-24574-4_28
4. Xu, Yongchao Géraud, Thierry Puybareau, Elodie Bloch, Isabelle Chazalon, Joseph. (2018). White Matter Hyperintensities Segmentation in a Few Seconds Using Fully Convolutional Network and Transfer Learning. https://doi.org/10.1007/978-3-319-75238-9_42

Iterative fully convolutional neural networks in the VerSe2019 challenge

Nikolas Lessmann

Diagnostic Image Analysis Group, Department of Radiology and Nuclear Medicine,
Radboud University Medical Center Nijmegen, The Netherlands
nikolas.lessmann@radboudumc.nl

1 Method

We participated in the VerSe2019 challenge with a previously published method based on iteratively applied fully convolutional neural networks [2]. Briefly, this method relies on a U-net-like 3D network that analyzes a $128 \times 128 \times 128$ voxel region of interest in the image. In this region of interest, the network segments and labels only the bottom-most visible vertebra and ignores other vertebrae that may be (partly) visible in the region of interest. The region of interest is iteratively moved over the image by moving it to the center of the detected piece of vertebra after each segmentation step. If only part of a vertebra was detected, moving the region of interest to the center of the detected fragment ensures that a larger part of the vertebra becomes visible for the next iteration. Once the entire vertebra is visible in the region of interest, the segmentation and labeling results are stored in a memory component. This memory is a binary mask that is an additional input to the network and is used by the network to recognize and ignore already segmented vertebrae. By repeating the process of searching for a piece of vertebra and following this piece until the whole vertebra is visible in the region of interest, all vertebrae are segmented and labeled one after the other. When the end of the scan is reached, the predicted labels of all detected vertebrae are combined in a global maximum likelihood model to determine a plausible labeling for the entire scan, thus avoiding duplicate labels or gaps. Please refer to [2] for further details.

This method has previously been trained and tested mainly with low-dose chest CT scans, and additionally with thoracolumbar and lumbar spine CT scans. However, these datasets were still much more homogeneous than the VerSe2019 challenge dataset. One of the lumbar CT datasets contained several compression fractures, but metal implants were not present in any of the scans. The method has also not been tested on the cervical spine before.

Our goal with this submission was therefore to test the published algorithm on a new dataset, making only few changes and keeping the underlying segmentation strategy the same. This algorithm was mainly focused at segmentation and we therefore made a few changes in an attempt to stabilize the labeling performance.

1.1 Training data

We manually removed spurious voxels from the reference annotations (such as L6 voxels in cervical spine scans). Furthermore, we cropped all training images at the center slice of the top-most and bottom-most annotated vertebrae to ensure that only annotated vertebrae are visible. Since our method works by searching for the very first visible vertebrae and then moves along the spine, having not annotated vertebrae in the training data would have posed a challenge. Consequently, our method therefore also segments all vertebrae visible in a scan, not only completely visible vertebrae. Originally, our method also predicted for each detected vertebra whether it is completely or incompletely visible in the scan. However, since in the VerSe2019 challenge only completely visible vertebrae were annotated, we did not use this output.

In addition to the challenge training set, we used two publicly available datasets for training: The dataset from the vertebra segmentation challenge at the Computational Spine Workshop (CSI) at MICCAI 2014[3] and the xVert-Seg.v1 dataset[1]. These are thoracolumbar and lumbar spine CT scans of both healthy adults and of adults with compression fractures.

1.2 Loss function

Compared with the published method [2], we made a few small modifications to the loss function: The anatomical labeling of detected vertebrae was optimized by minimizing a combination of L2 and L1 norm, in which the L1 norm was weighted ten times as much as than the L2 norm. The original paper relied on the L1 norm only. Similarly, we used a combination of the originally proposed segmentation error and the voxelwise categorical crossentropy as loss function for the segmentation output of the network. Furthermore, we presented more empty patches, with no vertebra voxels or no remaining unsegmented vertebra voxels, to the network: every third instead of every fourth patch was empty.

1.3 Rib detection

To improve the labeling accuracy, we trained a second network to predict whether a vertebra is a thoracic vertebra or not. This network receives as input the final image patch in which a vertebra is segmented and the corresponding segmentation mask as a second channel. The network has a simple architecture based on $3 \times 3 \times 3$ convolutions, batch normalization and max-pooling. The final layer is a dense layer with sigmoid activation function. At inference time, the first thoracic vertebra and the first cervical vertebra as identified by this auxiliary network had stronger influence on the label voting. Their vote counted three times as much as that of other vertebrae.

1.4 Cropping at inference time

At inference time, we added a step in which we crop the image along the z-axis in steps of 2.5 % from the bottom if no vertebra was found in the entire scan.

This helps in case the very first, i.e., bottom-most, vertebra is only visible with a very small fragment. This small element might be too small to be detected as vertebra, but might prevent the network from detecting any vertebra above as the bottom-most vertebra. If the first visible vertebra is not properly detected, the whole iterative process might fail.

1.5 Centroid estimation

We did not incorporate the vertebra centroids provided with the training data. However, we noticed that these do not correspond to the actual centroid of the vertebra as measured from the segmentation mask. Therefore, we estimated the offset between the centroids measured from the segmentation mask and the expected output. We used the reference segmentations to obtain a list of centroids \mathbf{v}_i with corresponding reference centroid coordinates \mathbf{w}_i . Subsequently, for each vertebra individually, we searched for an offset $\boldsymbol{\delta}$ by minimizing the following cost function:

$$\min_{\boldsymbol{\delta}} \sum_i \mathbf{v}_i - \mathbf{w}_i + \boldsymbol{\delta} \quad (1)$$

References

1. Ibragimov, B., Korez, R., Likar, B., Pernus, F., Xing, L., Vrtovec, T.: Segmentation of pathological structures by landmark-assisted deformable models. *IEEE Transactions on Medical Imaging* **36**(7), 1457–69 (2017). <https://doi.org/10.1109/tmi.2017.2667578>
2. Lessmann, N., van Ginneken, B., de Jong, P.A., Igum, I.: Iterative fully convolutional neural networks for automatic vertebra segmentation and identification. *Medical Image Analysis* **53**, 142–155 (2019). <https://doi.org/10.1016/j.media.2019.02.005>
3. Yao, J., Burns, J.E., Forsberg, D., Seitel, A., Rasoulian, A., Abolmaesumi, P., Hamernik, K., Urschler, M., Ibragimov, B., Korez, R., et al.: A multi-center milestone study of clinical vertebral CT segmentation. *Computerized Medical Imaging and Graphics* **49**, 16–28 (2016)

Vertebra Labeling and Segmentation in 3D CT using Deep Neural Networks

Dong Yang, Ziyue Xu, Xiaosong Wang, Qihang Yu, Holger Roth
, and Daguang Xu

NVIDIA

Abstract. We introduce an automated and efficient approach for vertebral segmentation and labeling in 3D CT. A U-shape deep neural network is used for generating the vertebral segmentation and labels. And the model ensemble and post-processing is further adopted to generate final prediction.

Keywords: Vertebra segmentation · Deep neural works.

1 Problem formulation

The problem is formulated as a 26-class segmentation task given 3D CT as input. The class information from prediction is able to provide labels (cervical $C1 \sim C7$, thoracic $T1 \sim T12$, lumbar $L1 \sim L6$) for different vertebrae. For vertebra localization, the centroids of vertebrae are determined as the mass centers of segmentation masks.

2 Methodology

We have adopted a U-shape neural network for vertebral segmentation following the fashion of the state-of-the-art network for 3D medical image segmentation [1–3]. The network architecture is nearly symmetric with an encoder and a decoder. And different operations (e.g. convolution, max-pooling, up-sampling, batch normalization, etc.) are contained inside the encoder and decoder to increase the receptive field. After achieving the segmentation results, the centroids of vertebrae are computed based on the mass centers of binary labels for each individual vertebra. To help determining the vertebral body center, several iterations of morphological erosion are conducted to remove the vertebral “wings”.

3 Experiments

We use the 5-fold cross validation on the training dataset. The neural network models are trained after 30,000 iterations using soft dice loss [2]. The learning rate is 0.0001. The validation metric is the Dice’s score. Then the final prediction is from the ensemble of five models.

References

1. Cicek, Ozgun, 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, pp. 424-432. Springer, Cham, 2016.
2. Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." In 2016 Fourth International Conference on 3D Vision (3DV), pp. 565-571. IEEE, 2016.
3. Myronenko, Andriy. "3D MRI brain tumor segmentation using autoencoder regularization." In International MICCAI Brainlesion Workshop, pp. 311-320. Springer, Cham, 2018.

Combining Template Matching with CNNs for Vertebra Segmentation, Localization and Identification

Report submitted to 2019 MICCAI VERSE* challenge

Felix Ambellan¹, Tamaz Amiranashvili^{1,3}, Moritz Ehlke², Hans Lamecker²,
Sebastian Lehnert², Marilia Lirio², Nicolás Pérez de Olaguer², Heiko Ramm²,
Manish Sahu¹, Alexander Tack^{1,2,3}, and Stefan Zachow^{1,2}

¹ Zuse Institute Berlin, Berlin, Germany

<https://www.zib.de/visual/therapy-planning>

² 1000shapes GmbH, Berlin, Germany

<https://www.1000shapes.com>

³ Corresponding authors: amiranashvili@zib.de, alexander.tack@1000shapes.com

1 Introduction

The purpose of this work is to label all vertebra in a given field-of-view of a Computed Tomography (CT) scan and to generate voxel-level segmentations of the vertebrae present in the scan. This task is to be performed in a fully automated manner, resulting in a number of challenges: (1) The scans exhibit varying fields-of-view, containing different numbers of vertebrae in various regions of the spine. (2) Scans are of different resolutions. (3) CTs can include implants or other foreign materials, as well as pathologies. (4) Neighboring vertebrae often have a similar appearance in the CTs, which renders unique identification a difficult task, especially when the spinal scans are truncated.

2 Method

We address these challenges by decomposing the tasks at hand into a number of steps (Fig. 1):

1. In the first stage, we compute a multi-label segmentation with arbitrary, but separate labels for each vertebra visible in the image. This is achieved based on local regions of interest in the image, containing only the segmented (target) vertebra and surrounding structures.
2. The second stage assigns unique labels to the voxel segmentations of individual vertebra based on their shape and appearance, while globally regularizing over the whole field-of-view of the CT.

* The Large Scale Vertebrae Segmentation Challenge (VerSe2019).
<https://verse2019.grand-challenge.org/Home/>

3. The third stage derives landmark positions from the multi-label segmentations by applying a shape-based approach.

In the following, the individual stages of the pipeline are described in more detail.

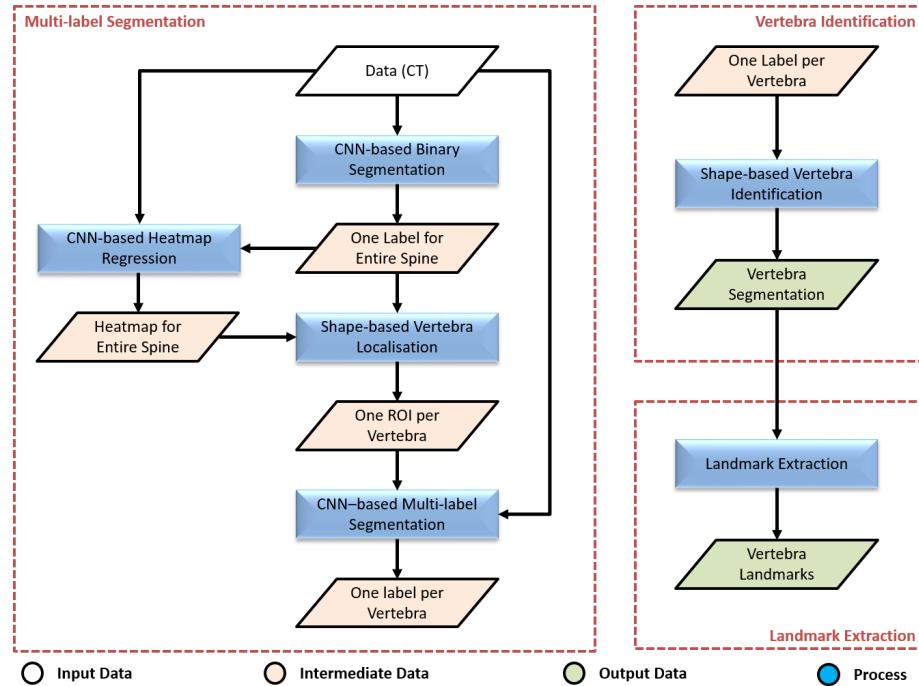


Fig. 1. Overview of the pipeline for segmentation, localization and identification of vertebrae. *Input:* CT dataset. *Output:* Voxel-level segmentation of vertebrae and detected 3D landmarks at the centers of the vertebral bodies.

2.1 Stage I: Multi-label Segmentation

The goal is to compute multi-label segmentation masks with arbitrary, but separate labels per vertebra. This includes processing steps for

- Creating a first, rough binary segmentation of the overall spine (*Binary Segmentation*)
- Localizing regions of interests around each vertebra (*Rough Vertebra Localization*)
- Performing voxel-level, high-quality segmentation of each vertebra (*Multi-label Segmentation*)

Binary Segmentation First, a binary segmentation is performed, separating the spine from the background. This is achieved through a UNet [3], which is employed on 2D sagittal slices. For each slice, neighboring slices are included as additional channels in the input to provide a larger context, following [1]. Being a fully convolutional network, the UNet is approximately translation invariant, making it especially suitable for working with images with varying fields-of-view and resolutions. Furthermore, the local nature of the receptive field of fully-convolutional networks suits the binary segmentation task, since global context is not required (in contrast to, for example, identification). The network is trained on fixed-size, random crops from original slices in order to allow training with batch sizes larger than one.

Rough Vertebra Localization In the next step, the number of vertebra as well as their rough positions are computed based on the binary segmentation mask. We achieve this by combining shape-based fitting via *generalized Hough transform* (GHT), following [4], with a CNN-based heat-map regression for localizing vertebra in the spinal column. The idea is to restrict the predicted locations of GHT templates to regions of interest regressed by a Butterfly CNN [2]. To deal with the variation in shape of the vertebra in different regions of the spinal column, we used manually generated GHT templates of the lumbar (L5-L1), lower thoracic (T12-T10), mid-thoracic (T9-T5), upper thoracic (T4-T1), lower-to-mid cervical (C5-C3), and upper cervical (C2-C1) spine in the fitting procedure. The Butterfly network was trained on mean and maximum intensity projections in anterior-posterior and lateral directions of the CTs (Phases 1-3), with target regions extracted from the ground-truth landmarks.

Multi-label Segmentation Based on the rough locations from the previous step, a region of interest (bounding box) is derived for each visible vertebra. Individual vertebrae are then segmented via a UNet [3] based on 2D sagittal slices, cropped to the corresponding regions of interests. Similar to the binary segmentation step, more context is provided by including neighboring slices as additional input channels. Since every image crop only contains a single fully visible vertebra, the network is able to separate the target vertebrae from the neighboring ones. The segmentations resulting from individual image crops are then combined into a multi-label segmentation mask, with one label for each vertebra visible in the image.

2.2 Stage II: Vertebra Identification

At this stage, we aim at identify types of individual vertebra in the image. Vertebra identification is performed based on shape through template fitting. In addition, explicit global regularization over the whole visible spine is employed to achieve robustness.

For each individual vertebra, shape templates are fitted non-rigidly to the given labels via *iterative closest points* (ICP) algorithm. Here, we again use six

templates (lumbar, lower thoracic, thoracic mid section, upper thoracic, lower-mid cervical, upper cervical) for the different groups of vertebrae. The result of the matching procedure is a table that contains a fitting score for each template and each detected label. We then optimize for the unique set of labels in the table that maximizes the combined score while maintaining consistent ordering of vertebra (e.g. L4 must follow L5). The multi-label segmentation of the previous stage is re-labeled according to the determined ordering, resulting in a segmentation with uniquely identified labels for each vertebra.

2.3 Stage III: Landmark Extraction

After segmentation masks and types have been extracted, we compute the landmark position by re-fitting a template of the vertebral body of each vertebra to the unique labels and by extracting the center point from the template. A unique landmark is then assigned to each center point for final output.

3 Implementation

The method was implemented using Tensorflow⁴ for deep learning and inference as well as in ZIBAmira⁵ for GHT and ICP fitting methods. Refer to Table 1 for more details on software libraries and tools used throughout the pipeline.

Table 1. List of third-party tools and libraries used for implementing the overall approach.

Name	Version
Python	3.5
Tensorflow	1.12.3
NumPy	1.16.4
SciPy	1.3.0
Pandas	0.23.0
Opencv-python	4.1.0.25
nibabel	2.4.1
ZIBAmira	2019 series

3.1 Data

In addition to the official challenge data, Dataset 2 [5] and Dataset 15 [6] from the SpineWeb Database⁶ were used for training of CNNs in the multi-label segmentation stage.

⁴ www.tensorflow.org

⁵ amira.zib.de

⁶ <http://spineweb.digitalimaginggroup.ca/spineweb/>

References

1. F. Ambellan, A. Tack, M. Ehlke, S. Zachow. *Automated Segmentation of Knee Bone and Cartilage combining Statistical Shape Knowledge and Convolutional Neural Networks: Data from the Osteoarthritis Initiative*. Medical Image Analysis, 52(2):109–118, 2019.
2. Y. Li, X. Cheng, and J. Lu. *Butterfly-Net: Optimal function representation based on convolutional neural networks*. Arxiv preprint, arXiv:1805.07451, 2018.
3. O. Ronneberger, P. Fischer, T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, in: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 246–253, 2015.
4. H. Seim, D. Kainmueller, M. Heller, H. Lamecker, S. Zachow, H.-C. Hege *Automatic Segmentation of the Pelvic Bones from CT Data Based on a Statistical Shape Model*, in: Eurographics Workshop on Visual Computing for Biomedicine (VCBM), pp. 93–100, 2015.
5. J. Yao, J. E. Burns, H. Muoz and R. M. Summers *Detection of Vertebral Body Fractures Based on Cortical Shell Unwrapping*, in: International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Volume 7512, pp 509–516, 2012.
6. J. Yao, J. E. Burns, D. Forsberg, A. Seitel, A. Rasoulian, P. Abolmaesumi, K. Hamernik, M. Urschler, B. Ibragimov, R. Korez, T. Vrtovec, I. Castro-Mateos, J. M. Pozo, A. F. Frangi, R. M. Summers, S. Li *A Multi-center Milestone Study of Clinical Vertebral CT Segmentation*, in: Computerized Medical Imaging and Graphics, 2016.