Neurospine 2024;21(1):57-67.
https://doi.org/10.14245/ns.2347178.589

Neurospine
pISSN 2586-6583 eISSN 2586-6591

Original Article

Whole Spine Segmentation Using Object Detection and Semantic Segmentation

Corresponding Author
Victor E. Staartjes
https://orcid.org/0000-0003-1039-2098

Machine Intelligence in Clinical Neuroscience (MICN) Laboratory, Department of Neurosurgery, Clinical Neuroscience Center, University Hospital Zürich, University of Zürich, Sternwartstrasse 6, Zürich 8091, Switzerland
Email: victoregon.staartjes@usz.ch

Received: November 1, 2023
Revised: January 6, 2024
Accepted: January 7, 2024

Raffaele Da Mutten[1], Olivier Zanier[1], Sven Theiler[1], Seung-Jun Ryu[2], Luca Regli[1], Carlo Serra[1], Victor E. Staartjes[1]

[1]Machine Intelligence in Clinical Neuroscience (MICN) Laboratory, Department of Neurosurgery, Clinical Neuroscience Center, University Hospital Zürich, University of Zürich, Zürich, Switzerland
[2]Department of Neurosurgery, Daejeon Eulji University Hospital, Eulji University Medical School, Daejeon, Korea

Objective: Virtual and augmented reality have enjoyed increased attention in spine surgery. Preoperative planning, pedicle screw placement, and surgical training are among the most studied use cases. Identifying osseous structures is a key aspect of navigating a 3-dimensional virtual reconstruction. To automate the otherwise time-consuming process of labeling vertebrae on each slice individually, we propose a fully automated pipeline that automates segmentation on computed tomography (CT) and which can form the basis for further virtual or augmented reality application and radiomic analysis.
Methods: Based on a large public dataset of annotated vertebral CT scans, we first trained a YOLOv8m (You-Only-Look-Once algorithm, Version 8 and size medium) to detect each vertebra individually. On the then cropped images, a 2D-U-Net was developed and externally validated on 2 different public datasets.
Results: Two hundred fourteen CT scans (cervical, thoracic, or lumbar spine) were used for model training, and 40 scans were used for external validation. Vertebra recognition achieved a mAP50 (mean average precision with Jaccard threshold of 0.5) of over 0.84, and the segmentation algorithm attained a mean Dice score of 0.75 ± 0.14 at internal, 0.77 ± 0.12 and 0.82 ± 0.14 at external validation, respectively.
Conclusion: We propose a 2-stage approach consisting of single vertebra labeling by an object detection algorithm followed by semantic segmentation. In our externally validated pilot study, we demonstrate robust performance for our object detection network in identifying individual vertebrae, as well as for our segmentation model in precisely delineating the bony structures.

Keywords: Machine learning, Deep learning, Spine, Artificial intelligence, Algorithms

INTRODUCTION

Virtual and augmented reality have enjoyed increased attention in spine surgery.[1-3] Preoperative planning, navigation for pedicle screw placement, and surgical training are among the most studied areas of application.[1] When navigating a 3-dimensional (3D) virtual reconstruction, identifying osseous structures on computed tomography (CT) is crucial. To automate the otherwise time-consuming process of labeling vertebrae on each individual slice, we propose a fully automated pipeline. Improved precision, decreased errors due to human fatigue, and increased consistency are suggested as benefits of incorporating automatic segmentation into clinical practice.[4,5] Also, virtual or augmented reality applications or radiomic analysis rely on target structure identification and segmentation.[6] Spine segmentation has been employed for disease diagnosis and preoperative treatment planning.[7,8] Augmented reality navigation has also been shown to improve the precision of screw insertion compared to free hand

approaches.[9] Exact delineations of target structures is necessary for this. Convolutional neural networks are increasingly used for medical image segmentation.[10]

U-Nets are frequently used for this task, and their efficacy has been sufficiently demonstrated.[11-14] Commonly, a region of interest needs to be manually defined first to further divide the image into anatomical regions, before segmentation can be performed.[4] By applying state-of-the-art object detection methods—trained on the task of creating bounding boxes around each individual vertebra—a greater level of automation and potentially enhanced segmentation precision could be achieved.[15,16]

This 2-stage approach enables training convolutional neural networks on slices with higher—or even native—resolution, potentially increasing vertebral detection precision. This so-called patch-wise segmentation has a variety of benefits, from improved memory efficiency to addressing class imbalance of small structures.[17] Various strategies for defining field of views are applied in the VerSe 20 Challenge.[18] For example, Chen et al.[18] use a 3D U-Net for localization by initially generating random patches and then use these predictions to crop precisely. Payer et al.[19] generate heatmaps of the spine, identify the centroid of the vertebrae and crop a 3D patch around it. Yet, no attempt at using You-Only-Look-Once (YOLO) algorithms for patch-generation has been made. This allows for adaptive patch size around the precise edges of the vertebra. In summary, in this pilot study, we evaluate the feasibility of segmenting CTs of the cervical, thoracic, or lumbar spine using the proposed 2-stage machine learning approach with object detection followed by semantic segmentation.

MATERIALS AND METHODS

1. Data Collection and Preprocessing

We utilized 3 datasets: one for training and a subset of 2 others for external evaluation.

The dataset used for training is publicly available (VerSe 20 Challenge) and consists of 214 patients from a variety of centers and vendors (Siemens, GE, Philips, and Toshiba). The dataset had the following inclusion criteria: Minimum age of 18, 7 fully visualized vertebrae (without counting sacral od transitional vertebrae) and minimum pixel spacing of 1.5 mm (craniocaudal), 1 mm (anterior-posterior), 3 mm (left-right).[20] Exclusion criteria were traumatic fractures and bony metastases.[18,20-22] The respective labels (26 different labels for the vertebrae from C1 to L5) were created through a semiautomated process: Initially, suggestions were made by an algorithm, which were subsequently refined by human experts.[18,20,21] Two medical students, specifically trained on the task, manually corrected the suggestion by the algorithm in a laborious process. This was performed in the original image space. The labels were subsequently validated by a neuroradiologist. Since the suggested segmentation mask was a 3D volume, we assume that all slice directions were considered in subsequent refinement process. These labels incorporated within the VerSe 20 dataset were used as ground truth for model training. For training purposes, this dataset consisting of 214 patients was split into 173 for training and validation during training (on a random 20% of the images) and a holdout dataset of 41 patients for internal validation.

Subsequently, to evaluate the out-of-sample performance of our fully trained method, we chose to test its performance on 2 other unrelated datasets: for the first external evaluation, we used a COVID-19 dataset (CT images in COVID-19) consisting of chest CTs captured at the initial point of care[23-25] that show the thoracic spine. Second, we evaluated our pipeline on 20 liver CTs, which span the lumbar and thoracic spinal regions and have also been part of a semantic segmentation challenge called Medical Segmentation Decathlon, of which we used 1 of the 10 subsets (MSD T10).[26,27] The corresponding ground-truth labels were obtained from the CT1kSpine dataset,[28] that created them in a semiautomated fashion from a nnU-Net that was updated every 100 cases. We opted for these 2 external validation datasets for the following reasons. First, they were suitable for our purposes since they were publicly available. Second, the manual segmentations came from the same dataset as the labels for training. Third, we wanted to evaluate the robustness of our approach on CTs that were not centered on the spine.

2. Preprocessing

We resampled the voxel size to isotropic $1.0 \times 1.0 \times 1.0$ and padded the images to be uniform in size for all dimensions. Intensities were windowed with a center of 300 and a range of 2,000 and normalized with respect to their minimum and maximum.

3. Model Development

The training process is visually depicted in Fig. 1. First, a YOLO algorithm, Version 8 and size medium, (YOLOv8m) was trained to regions of interest.[29,30] Similar to cars detecting pedestrians,[31] this learned to identify each vertebral body level and to create a bounding box for this. These regions were then cropped to the box size with a small margin of 5 pixels to ensure all corners of the vertebrae were on the smaller image. After cropping, these

https://doi.org/10.14245/ns.2347178.589

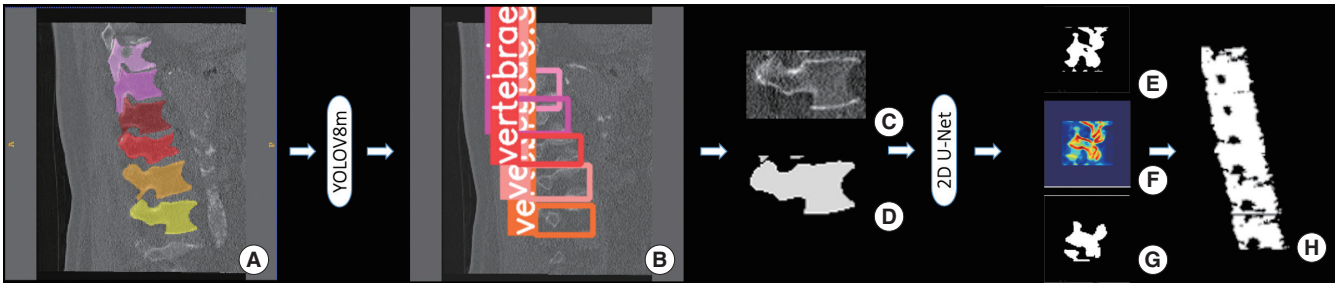

Fig. 1. An exemplary illustration of our pipeline is shown. CT slice as input is used for object detection, then cropped and a 2D-U-net for segmentation is trained and evaluated. (A) Input image with manual segmentations. (B) Object detections on CT before cropping. (C) Cropped input image for U-Net. (D) Cropped input mask for U-Net. (E) Thresholded prediction of U-Net. (F) Probability map generated from U-Net. (G) Cropped Segmentation to compare U-Net performance. (H) The cropped predictions are reassembled into a full segmentation. 2D, 2-dimensional.

extracted regions were resized to 256×256 and used as input for 2D-U-Net training.[11] As a convolutional network, it extracts feature maps in the contracting path. After a rectified linear unit activation function introduces nonlinearity, a max pooling operation takes the highest feature from each two-by-two square to half the dimensions. Subsequently, the expansion path uses transposed convolutions to reverse the reduction in dimension of the contracting path. Concatenation of feature maps on each symmetrical level helps restore spatial information. Finally, a sigmoid activation function assigns pixel-wise values to the segmentation mask. Due to this reduction and expansion in size, the architecture can be visualized in the shape of a "U," hence the name of the model architecture. The following platforms were used: Python 3.9.0,[32] Keras 2.5.0,[33] SimpleITK,[34] and nibabel.[35] The training was conducted on a Nvidia RTX 3090 graphical processing unit (GPU).

As a result of hyperparameter tuning, the best-performing model was trained for 14 epochs using early stopping. The final U-Net architecture consisted of 96 starting neurons, a depth of 3 with 4 blocks on each level. Binary cross-entropy was used as loss function, and a batch size of 80 yielded the best performance.

4. Evaluation

Precision (positive predictive value), recall (sensitivity), and mean average precision (mAP) with a Jaccard threshold of 0.5 (mAP50) were assessed as standard bounding-box evaluation metrics.[36] Furthermore, mAP50-95, which corresponds to mAP with 10 Jaccard score steps from 0.5–0.95 with steps of 0.05, was implemented. In those, a box reaching the respective predefined threshold is considered to be a true positive. To put this into perspective, one of the best values for a benchmark dataset called COCO are 0.79 for mAP50 and 0.66 for mAP50-95.[37]

U-Net performance was determined by comparing Dice score, Jaccard score and the 95th percentile of the Hausdorff distance of labels and predictions.[38-41] Dice and Jaccard score are measures of overlap ranging from zero—indicating no congruence—to one for a perfect match. While the Dice score is defined as twice the area of overlap divided by the sum of both areas, the Jaccard score describes the intersection divided by the union. Both metrics are ultimately a quotient of the correctly classified region and the ground-truth mask. The Hausdorff distance analyses the distance between 2 sets of points that are derived from the edges of the segmentations.

Evaluation was performed on the held-out VerSe 20 data, as well as the 2 external validation datasets. Mean and standard deviation as well as median and interquartile range are reported where appropriate.

RESULTS

1. Cohort

A total of 173 CT scans were used for training, 41 for internal validation, and 20 for each of the 2 external validation sets. Patient and radiological information, as reported by the respective datasets, is summarized in Table 1. The training dataset consisted of (mean ± standard deviation [SD]), 523 ± 48 coronary, 600 ± 267 axial, and 537 ± 358 sagittal slices. The first external validation set with liver scans was comprised of 533 ± 71 coronary, 512 ± 96 axial, and 533 ± 71 sagittal slices and the second, chest CT dataset, entailed 400 ± 57 slices in all dimensions. The entire VerSe 20 dataset with 300 patients (86 from 2019 with lower resolution and 214 new cases) consists of 144 female patients (48%), with a (mean ± SD) age of 56.2 ± 17.6. A total of 4,142 (100%) individual vertebrae were labeled, of which 581 (14%) were cervical, 2,255 (54%) thoracic and 1,306 (32%) lumbar vertebrae.

Table 1. Summary of the patient and radiological characteristics

Variable	Dataset		
	VerSe	MSD T10	COVID-19
CT region	Spine	Liver	Chest
Baseline			
No. of patients	214	20	20
Age (yr)	59.00 ± 17.00	NA	NA
Voxel dimensions			
Pixel spacing (mm)	0.34 ± 0.16	0.95 ± 0.11	1 ± 0
Slice thickness (mm)	1.24 ± 0.06	1.04 ± 0.11	1 ± 0

Values are presented as mean ± standard deviation.
CT, computed tomography; NA, not available.
Voxel dimensions were only available for the entire respective dataset.

2. Object Detection: Identifying and Labeling Each Vertebral Level

A pretrained YOLOv8m with over 25 million parameters was trained on 40'446 images for 57 epochs using an early stopping function. Evaluation of the training, internal validation (holdout), and pool external validation performance resulted in a mAP50-95 of 0.64, 0.63, and 0.09 across all classes. Table 2 depicts detailed results per class for all the evaluated datasets. Precision and recall for different confidence levels are shown in Fig. 2. Inference time per slice for training, internal validation and pool external validation was 3.2 msec, 3.2 msec, and 5 msec, respectively.

3. Semantic Segmentation: Delineating Bony Structures

After object detection, single-vertebral-level cropped images were used to train and validate a semantic segmentation network: 94'184 cropped images were used for training. Training, internal validation (holdout), and pooled external validation showed a mean Dice of 0.75 ± 0.14, 0.76 ± 0.12, and 0.79 ± 0.1695, respectively. Detailed results are shown in Table 3. The distribution of the metrics is presented in Fig. 3. Exemplary results are depicted in Fig. 4. Following prediction, the cropped slices were reassembled into 3 dimensions. Inference time per slice for training, internal validation and pool external validation was 68 msec, 51 msec, and 44 msec.

DISCUSSION

We have developed and validated a pipeline for automated whole spine anatomical segmentation combining YOLO object detection and a 2D-U-Net for subsequent semantic segmentation. Generalizability was demonstrated by evaluating the performance on 2 different datasets for external validation. Our object detection method showed robust performance. By setting a low confidence threshold for detection, the risk of missing out on slices to segment is minimized. Segmentation was observed to yield robust results as well. At external validation, evaluation of our entire pipeline (semantic segmentation results) demonstrated robustness without signs of overfitting, as segmentation performance (incorporating the object detection method as a preprocessing step) was equal to training set performance.

Segmentation of medical imaging has become one of the major fields of clinical machine learning for many reasons within the last decade: Applications such as automated diagnostics and volumetric measurements, radiomics, and generation of virtual or augmented reality visualizations for demonstration, surgical training, and intraoperative navigation all necessitate robust methods for segmenting anatomical structures from native x-ray, CT or magnetic resonance imaging data.[42-44] However, the process of manually segmenting images—or manually correcting pre-segmented or thresholded images in the sense of semiautomated approaches—is highly time-consuming and would render broad adoption of the abovementioned applications into the clinical routine impossible.[45]

Machine learning methods have thus markedly helped to cut down on time and effort needed for creating segmentations of medical images: For example, subarachnoid blood, intra-axial brain tumors, or pituitary adenomas can be readily segmented in a fully automated approach, in a time-efficient manner.[46-48]

Normally, medical images are preprocessed and directly segmented by an algorithm, or specific regions of interest have to be manually delineated before segmentation, which is resource intensive and a potential source of errors. The spine is a large anatomical structure with clearly delineated segments (vertebral levels) and – at the bony level – regular interruptions (disc spaces), which would theoretically enable the use of object recognition algorithms to parcellate the spine into multiple smaller structures. Those extracted subregions can then be fed into algorithms with native or near-native resolution for a more precise and computationally efficient bony structure delineation. In addition, the obvious added benefit of object detection here is that vertebral levels are automatically recognized and labeled (for example, C7). From a generated segmentation, numerous parameters can be extracted. Hohn et al.[49] used a simpler thresholding approach combined with manual segmentation of subregions to determine bone quality. Maintaining high resolution data of the initial CT only helps generate more accurate estimations of bone mineral density. Siemionow et al.[50] assessed vari-

https://doi.org/10.14245/ns.2347178.589

Table 2. Yolov8m performance during training and on the holdout sets

Segment	Instances				Precision				Recall				mAP50				mAP50-95			
	Val	Hold	Ext1	Ext2	Val	Hold	Ext1	Ext2	Val	Hold	Ext1	Ext2	Val	Hold	Ext1	Ext2	Val	Hold	Ext1	Ext2
All	16,634	16,351	10,862	13,482	0.899	0.906	0.780	0.235	0.780	0.775	0.347	0.155	0.849	0.845	0.405	0.130	0.638	0.632	0.145	0.038
C1	224	243	0	0	0.955	0.936	-	-	0.897	0.827	-	-	0.932	0.889	-	-	0.748	0.691	-	-
C2	181	180	0	0	0.909	0.931	-	-	0.818	0.823	-	-	0.861	0.861	-	-	0.664	0.644	-	-
C3	180	177	0	0	0.900	0.919	-	-	0.822	0.764	-	-	0.877	0.827	-	-	0.673	0.636	-	-
C4	184	177	0	0	0.924	0.946	-	-	0.793	0.831	-	-	0.848	0.882	-	-	0.660	0.683	-	-
C5	198	196	0	0	0.953	0.923	-	-	0.827	0.792	-	-	0.886	0.871	-	-	0.677	0.667	-	-
C6	216	218	0	46	0.836	0.870	-	0	0.810	0.784	-	0	0.856	0.821	-	0	0.645	0.639	-	0
C7	384	385	0	236	0.804	0.764	-	0	0.703	0.649	-	0	0.763	0.713	-	0	0.513	0.497	-	0
Th1	652	684	0	560	0.815	0.822	-	0.112	0.721	0.711	-	0.030	0.780	0.78	-	0.021	0.511	0.516	-	0.009
Th2	636	674	0	996	0.825	0.842	-	0.185	0.701	0.727	-	0.056	0.784	0.803	-	0.032	0.506	0.522	-	0.008
Th3	498	524	0	1,192	0.847	0.861	-	0.148	0.679	0.677	-	0.055	0.778	0.788	-	0.043	0.501	0.500	-	0.011
Th4	496	515	0	1,195	0.910	0.897	-	0.155	0.677	0.680	-	0.103	0.773	0.801	-	0.072	0.522	0.508	-	0.016
Th5	505	521	0	1,211	0.911	0.915	-	0.205	0.673	0.702	-	0.162	0.806	0.822	-	0.107	0.533	0.556	-	0.021
Th6	508	528	0	1,215	0.890	0.915	-	0.231	0.719	0.693	-	0.191	0.808	0.798	-	0.107	0.552	0.548	-	0.024
Th7	531	539	8	1,186	0.891	0.920	1	0.294	0.746	0.748	0	0.241	0.825	0.831	0	0.159	0.581	0.590	0	0.034
Th8	560	555	101	1,169	0.862	0.873	1	0.363	0.732	0.757	0	0.287	0.822	0.838	0.004	0.225	0.595	0.603	0.002	0.056
Th9	700	695	327	1,152	0.882	0.907	0.354	0.391	0.787	0.816	0.089	0.302	0.860	0.890	0.093	0.267	0.653	0.666	0.021	0.074
Th10	754	733	375	1,089	0.899	0.925	0.623	0.511	0.820	0.806	0.251	0.389	0.884	0.882	0.304	0.358	0.695	0.683	0.060	0.110
Th11	731	691	358	1,060	0.947	0.941	0.889	0.537	0.860	0.849	0.580	0.318	0.910	0.899	0.680	0.358	0.745	0.728	0.134	0.121
Th12	776	739	788	856	0.945	0.950	0.849	0.389	0.893	0.898	0.293	0.185	0.921	0.924	0.350	0.186	0.767	0.754	0.090	0.080
L1	1,322	1,248	1,471	319	0.922	0.913	0.770	0	0.795	0.778	0.474	0	0.859	0.854	0.533	0.011	0.697	0.682	0.199	0.005
L2	1,480	1,393	1,721	0	0.929	0.947	0.782	-	0.761	0.760	0.518	-	0.838	0.845	0.619	-	0.663	0.677	0.263	-
L3	1,623	1,567	1,895	0	0.936	0.925	0.750	-	0.760	0.751	0.536	-	0.843	0.838	0.606	-	0.670	0.662	0.256	-
L4	1,549	1,490	1,835	0	0.921	0.933	0.816	-	0.759	0.760	0.563	-	0.832	0.828	0.656	-	0.657	0.657	0.298	-
L5	1,665	1,598	1,983	0	0.913	0.909	0.748	-	0.846	0.832	0.509	-	0.904	0.884	0.611	-	0.722	0.700	0.275	-

YOLOv8m, You-Only-Look-Once algorithm, Version 8 and size medium; mAP50, mean average precision (mAP) with Jaccard threshold of 0.5; mAP50-95, mAP with threshold steps of 0.05 between 0.5 and 0.95; Val, validation performance during training; Hold, holdout performance of the Verse20 set; Ext1, external validation on the MSD T10 liver scans; Ext2, external validation on coronavirus disease 2019 chest computed tomography.

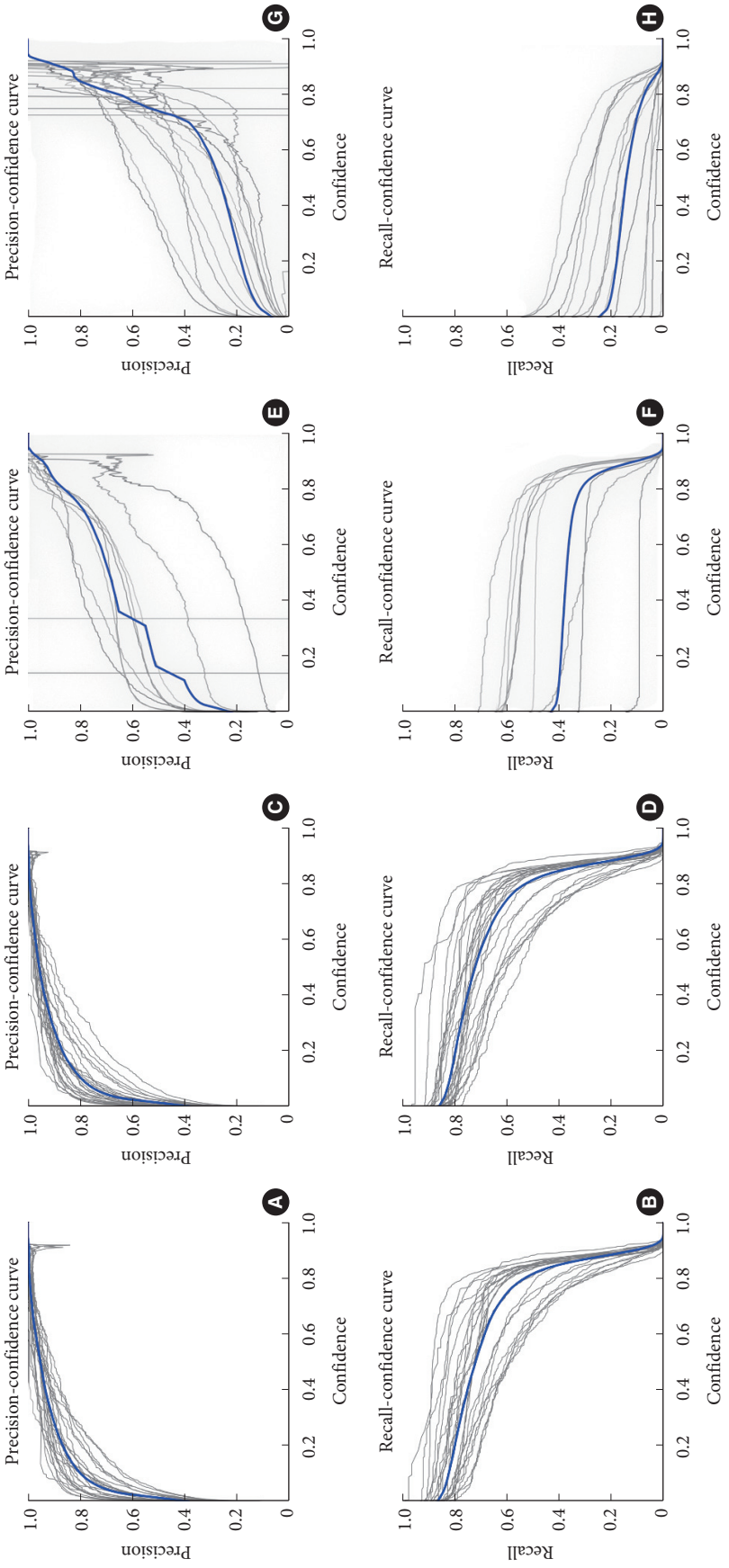

Fig. 2. Precision versus confidence plots of the YOLOv8m network, the blue line depicting performance across all classes: (A) training performance on VerSe 20, (C) holdout on VerSe 20, (E) MSD T10, (G) COVID-19. Recall versus confidence curves, the blue line depicting performance across all classes: (B) training performance on VerSe 20, (D) holdout on VerSe 20, (F) MSD T10, (H) COVID-19.

https://doi.org/10.14245/ns.2347178.589

Table 3. Performance of the U-Nets during both training and on held-out data

Variable	Dataset			
	VerSe 20		MSD T10	COVID-19
Performance type	Validation	Holdout	External validation	External validation
Dice				
Mean ± SD	0.750 ± 0.137	0.759 ± 0.119	0.770 ± 0.197	0.821 ± 0.142
Median (IQR)	0.793 (0.122)	0.796 (0.128)	0.829 (0.127)	0.861 (0.110)
Jaccard				
Mean ± SD	0.615 ± 0.144	0.624 ± 0.134	0.656 ± 0.192	0.715 ± 0.142
Median (IQR)	0.657 (0.162)	0.661 (0.171)	0.708 (0.181)	0.756 (0.168)
95th Percentile Hausdorff distance				
Mean ± SD	12.941 ± 12.346	12.383 ± 10.486	20.810 ± 14.604	22.832 ± 20.868
Median (IQR)	8.062 (7.770)	8.000 (7.597)	18.000 (12.820)	18.028 (27.053)

The metrics of both external validation sets are shown.
SD, standard deviation; IQR, interquartile range.

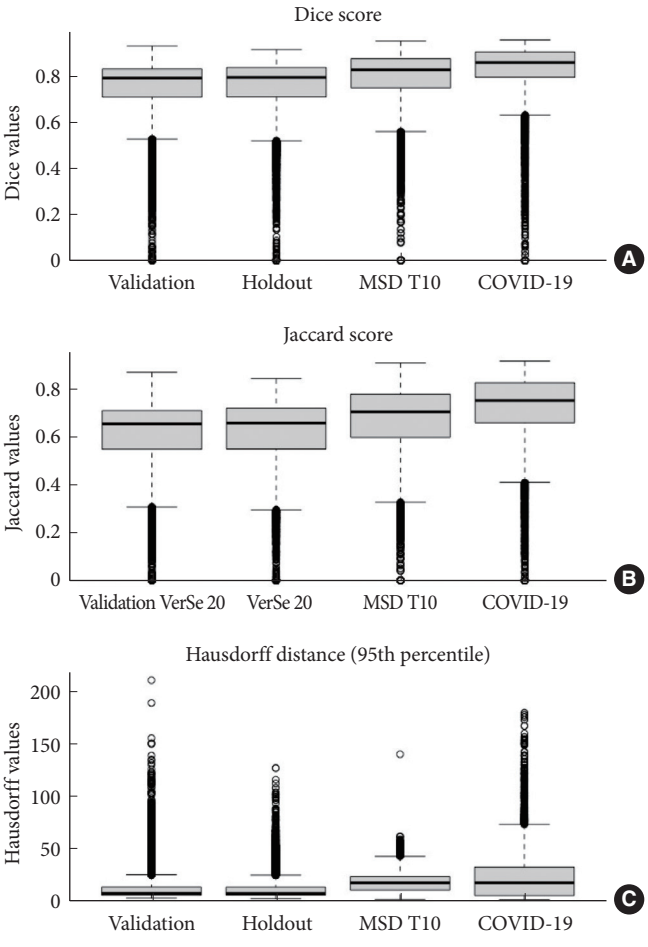

Fig. 3. Boxplots across all 4 evaluation sets: (A) Dice score, (B) Jaccard scroe, (C) 95th percentile Hausdorff distance.

ous parameters from segmenting spinal subregions. Those include vertebral body width, spinous process height, pedicle angulation and diameter at the isthmus. This provides the basis for automated operative planning, potentially assisting novice surgeons or reducing time needed for preoperative planning. These 2 examples not only illustrate the potential applications of automated spine segmentation, but also show the importance of high resolution patches without downsampling of image resolution.

We present a pilot study evaluating the feasibility and preliminary results of applying such a 2-stage automated segmentation pipeline, and generally shows that the concept is feasible and generalizes well to new images. Importantly, the external validation performance of the entire pipeline (semantic segmentation) was excellent—while the external validation performance of the object detection method itself seemed slightly less robust, which is partially inherent to the mAP metric. It is biased by model confidence, which tends to be lower during evaluation and, therefore, is more likely to fall below our predefined threshold. This trend can be seen in Fig. 2: External validation performance at a predetermined low confidence is inferior. Model confidence will be lower on new datasets, especially datasets that vary greatly from the training data, such as the liver and chest CTs used for external validation. Lower confidences do not necessarily equal worse bounding box generations, and the fact that final segmentation performance during external validation (building on the bounding boxes generated by our object detection algorithm) was still excellent is another indicator that our approach appears robust. Previous work with different approaches has yielded similar or even higher segmentation performance

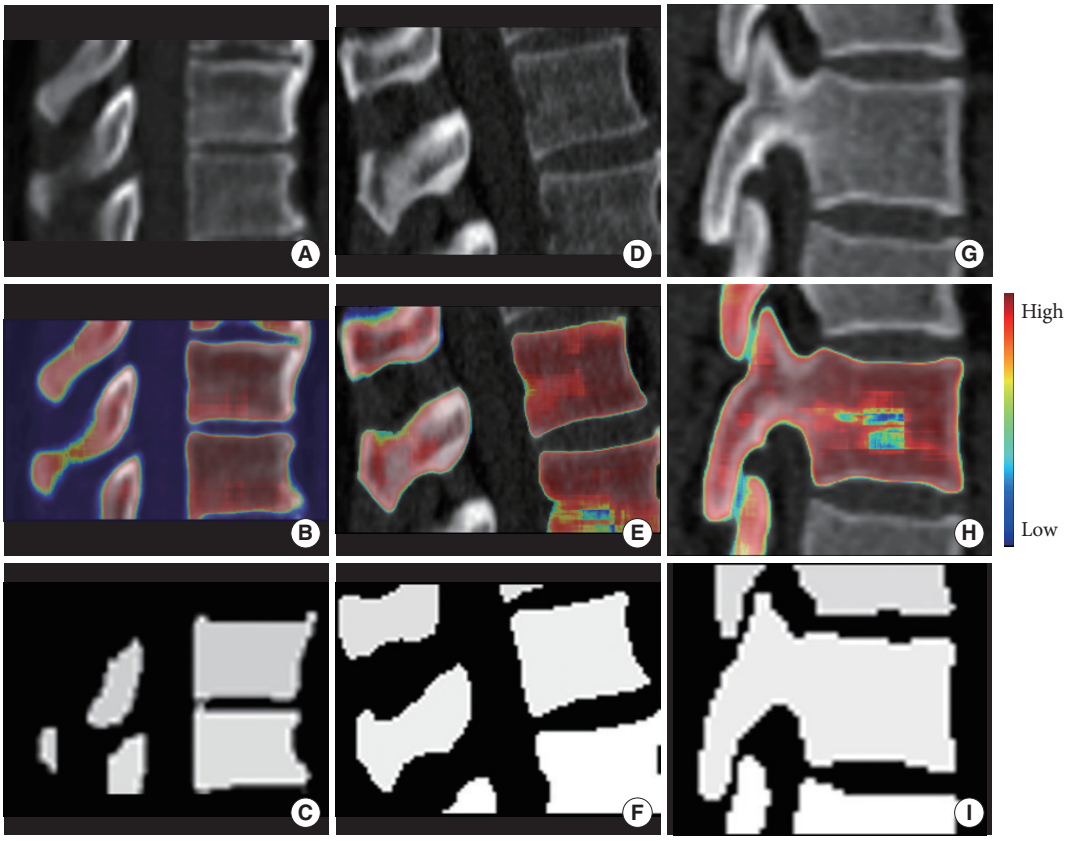

Fig. 4. Exemplary results from external validation set. (A) CT scan from VerSe 20 holdout set. (B) A with overlay of predicted mask; red signifies high probability, blue low. (C) Ground truth to A. (D) CT scan from the MSD 10 dataset. (E) D with predictions overlay. (F) Ground truth to D. (G) CT from the COVID-19 set. (H) G with prediction overlay; red signifies high probability, blue low. (I) Ground truth to G.

metrics.[18] This can mostly be attributed to 2 factors: First, we used rigorous 2-dimensional (2D) metrics since we performed our training on slices and not 3D volumes. On average, this will result in inferior performance (compared to e.g., 3D metrics), especially since we sometimes cropped segmentations made up only of a few pixels, and generally evaluated small volumes (spinal segments individually). One wrong pixel consequently has a bigger influence on measures of overlap compared to a full spinal imaging volume. This can be observed in Fig. 2 where the variability in metric performance becomes apparent. Secondly, our goal was to develop a clinically usable pipeline positioned at the optimal trade-off point between larger models' demand for more computational resources while not compromising on performance.[30] Lastly, we aimed to develop one approach for the whole spine, whereas many other models are focused solely on a specific subregion of the spine.

As stated in the proceeding of the VerSe Challenge,[18] a clinical spine CT scan is too large for GPU memory, and also for inference on clinical workstations with no designated GPU. Thus, either the resolution needs to be downsampled or the initial scan needs to be broken down into smaller pieces. As mentioned before, various techniques for this problem have been applied. We attempted to address this by a new, multistaged approach making use of the benefits of different models. The YOLO algorithms are largely used for real-time object detection, for example in autonomous driving where they are appreciated for their speed and precision.[31] U-Nets have been widely established in the field of biomedical image analysis.[14] Their main strength lies in precise segmentation, not detecting multiple objects on a large image. With 2D slices and cropping to a small portion of the initial image, we are able to drastically reduce the memory requirements of the input of the U-Net. For this pilot study, we also reduced image resolution for training, yet this is not a requirement for interference. With the implementation of a spatial pyramid pooling at the end of the YOLO architecture different input image scales can be efficiently handled.[29,51] For future inference, the CT could be used in native resolution for detection by the YOLO algorithm, and then cropped to the small patches. In then still

https://doi.org/10.14245/ns.2347178.589

native pixel scaling, the U-Net can be applied. In summary, we took the advantages of both models and combined them for optimal performance with reduced computational requirements.

With an ever increasing number of imaging studies carried out, automated supportive approaches such as ours can be of assistance in clinical practice,[43] and accurate segmentation has the potential to impact clinical practice and efficiency. With inference times per slice of under 0.1 second for both models combined, they are well below the time manual segmentation requires. While pre- and postprocessing, using a non-GPU environment and processing all slices per series requires more time, human input time is minimized extensively. Also, our approach paves the way for accurate labeling and segmentation of more detailed anatomical structures such as foramina, articular processes, facets, laminae, and pedicles.

Even though segmentation performance was good at external validation, not the entire spectrum of real-world CT data, spine anatomy, and pathology are represented by the available datasets. Larger and more heterogeneous datasets would be advantageous to cover more variation. For two of the datasets that are fully anonymized, information on patients' ages are not available, potentially limiting the generalizability of our results in different spine ages, for example in pediatric or geriatric patients. Also, training was partially confined in terms of model size and image resolution by computational constraints. Yet, smaller models usually have shorter inference times, which benefits potential applications to clinical routine on workstations without designated graphical processing units for machine learning. Equally, the image resolution had to be reduced for processing, resulting in pixelated masks if only a small region was cropped and then resized. This problem is inherent to our approach but could be reduced if training and predicting with higher pixel densities was less resource intensive. Finally, some regions of interest can be lost even with the high precision of our bounding box algorithm. Interpolating missing slices in postprocessing is feasible yet leads to a decrease in precision. Since we only trained on sagittal slices, mainly those are well segmented in 3 dimensions. Including slices from all dimensions and averaging the 3D predictions could reduce this effect in future work. Additionally, it could be hypothesized that more optimal, yet computationally demanding results could be achieved by using a 3D-U-Net. In order to minimize the input size, the mask generated by the YOLO algorithm could be applied to form a 3D volume, with the strategy described in our pilot study.

CONCLUSION

We propose a two-stage approach consisting of single vertebra labeling by an object detection algorithm followed by semantic segmentation. In our pilot study, including external validation, we demonstrate robust performance of our object detection network in identifying each vertebra individually, as well as for our segmentation model in exactly delineating the bony structures.

NOTES

Conflict of Interest: The authors have nothing to disclose.

Funding/Support: This study received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgments: We thank Massimo Bottini and Daniel de Wilde for their support in data preparation.

Author Contribution: Conceptualization: RDM, SJR, LR, CS, VES; Data curation : RDM, ST, OZ; Formal Analysis: RDM, VES, OZ; Investigation: RDM, VES, OZ; Methodology: RDM, VES, OZ; Project Administration: RDM, OZ, VES; Writing - Original Draft: RDM, VES; Writing - Review & Editing: RDM, OZ, ST, SJR, LR, CS, VES.

ORCID

Raffaele Da Mutten: 0000-0002-2645-0865
Olivier Zanier: 0000-0002-3286-3770
Sven Theiler: 0009-0008-0316-9799
Seung-Jun Ryu: 0000-0001-5547-9811
Luca Regli: 0000-0003-4639-4474
Carlo Serra: 0000-0002-7305-550X
Victor E. Staartjes: 0000-0003-1039-2098

REFRENCES

1. McCloskey K, Turlip R, Ahmad HS, et al. Virtual and augmented reality in spine surgery: a systematic review. World Neurosurg 2023;173:96-107.
2. Mishra R, Narayanan MDK, Umana GE, et al. Virtual Reality in neurosurgery: beyond neurosurgical planning. Int J Environ Res Public Health 2022;19:1719.
3. Ghaednia H, Fourman MS, Lans A, et al. Augmented and virtual reality in spine surgery, current applications and future potentials. Spine J 2021;21:1617-25.
4. Alzahrani Y, Boufama B. Biomedical image segmentation: a

survey. SN Comput Sci 2021;2:310.
5. Mharib AM, Ramli AR, Mashohor S, et al. Survey on liver CT image segmentation methods. Artif Intell Rev 2012;37:83-95.
6. Varkarakis V, Bazrafkan S, Corcoran P. Deep neural network and data augmentation methodology for off-axis iris segmentation in wearable headsets. Neural Netw 2020;121:101-21.
7. Han Z, Wei B, Mercado A, et al. Spine-GAN: semantic segmentation of multiple spinal structures. Med Image Anal 2018:50:23-35.
8. Molina CA, Phillips FM, Colman MW, et al. A cadaveric precision and accuracy analysis of augmented reality-mediated percutaneous pedicle implant insertion. J Neurosurg Spine 2020;34:316-24.
9. Dennler C, Safa NA, Bauer DE, et al. Augmented reality navigated sacral-alar-iliac screw insertion. Int J Spine Surg 2021;15:161-8.
10. King BF. Artificial intelligence and radiology: what will the future hold? J Am Coll Radiol 2018;15(3 Pt B):501-3.
11. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. arXiv:150504597 [Preprint]. 2015 [cited 2021 Nov 3]. Available from: http://arxiv.org/abs/1505.04597.
12. Bhalodiya JM, Lim Choi Keung SN, Arvanitis TN. Magnetic resonance image-based brain tumour segmentation methods: a systematic review. Digit Health 2022;8:20552076221074122.
13. Siddique N, Paheding S, Elkin CP, et al. U-Net and its variants for medical image segmentation: a review of theory and applications. IEEE Access 2021;9:82031-57.
14. Azad R, Aghdam EK, Rauland A, et al. Medical image segmentation review: the success of U-Net. arXiv:2211.14830v1 [Preprint]. 2022 [cited 2023 Oct 16]. Available from: https://arxiv.org/abs/2211.14830.
15. Jocher G, Chaurasia A, Qiu J. Ultralytics YOLOv8 [Internet]. GitHub, Inc.; 2024 [cited 2023 Oct 16]. Available from: https://github.com/ultralytics/ultralytics.
16. Terven J, Cordova-Esparza D. A comprehensive review of YOLO: from YOLOv1 and beyond. arXiv:2304.00501v7 [Preprint]. 2023 [cited 2023 Oct 16]. Available from: http://arxiv.org/abs/2304.00501.
17. Lee B, Yamanakkanavar N, Choi JY. Automatic segmentation of brain MRI using a novel patch-wise U-net deep architecture. PLoS One 2020;15:e0236493.
18. Sekuboyina A, Husseini ME, Bayat A, et al. VerSe: a vertebrae labelling and segmentation benchmark for multi-detector CT images. Med Image Anal 2021;73:102166.
19. Payer C, Štern D, Bischof H, et al. Coarse to fine vertebrae localization and segmentation with SpatialConfiguration-Net and U-Net. Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications; 2020 Feb 27-29. Valletta, Molta. SCITEPRESS - Science and Technology Publications; 2020:124-33. Available from: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0008975201240133.
20. Liebl H, Schinz D, Sekuboyina A, et al. A computed tomography vertebral segmentation dataset with anatomical variations and multi-vendor scanner data. arXiv:2103.06360v1 [Preprint]. 2021 [cited 2023 Oct 24]. Available from: http://arxiv.org/abs/2103.06360.
21. Löffler MT, Sekuboyina A, Jacob A, et al. A vertebral segmentation dataset with fracture grading. Radiol Artif Intell 2020;2:e190138.
22. Liebl H, Schinz D, Sekuboyina A, et al. A computed tomography vertebral segmentation dataset with anatomical variations and multi-vendor scanner data. Sci Data 2021;8:284.
23. An P, Xu S, Harmon SA, et al. CT Images in COVID-19 [Internet]. The Cancer Imaging Archive; 2020 [cited 2023 Oct 24]. Available from: https://wiki.cancerimagingarchive.net/x/o5QvB.
24. Harmon SA, Sanford TH, Xu S, et al. Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. Nat Commun 2020;11:4080.
25. Clark K, Vendt B, Smith K, et al. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. J Digit Imaging 2013;26:1045-57.
26. Antonelli M, Reinke A, Bakas S, et al. The medical segmentation decathlon. Nat Commun 2022;13:4128.
27. Simpson AL, Antonelli M, Bakas S, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv:1902.09063v1 [Preprint]. 2019 [cited 2023 Oct 10]. Available from: https://arxiv.org/abs/1902.09063.
28. Deng Y, Wang C, Hui Y, et al. CTSpine1K: a large-scale dataset for spinal vertebrae segmentation in computed tomography. arXiv:2105.14711v3 [Preprint]. 2021 [cited 2023 Oct 10]. Available from: https://arxiv.org/abs/2105.14711.
29. Jocher G, Changyu L, Hogan A, et al. ultralytics/yolov5: initial release [Internet]. Geneva (Switzerland): CERN; 2020 [cited 2023 Oct 10]. Available from: https://zenodo.org/record/3908560.
30. Redmon J, Divvala S, Girshick R, et al. You Only Look Once: unified, real-time object detection. arXiv:1506.02640v5

 https://doi.org/10.14245/ns.2347178.589

[Preprint]. 2015 [cited 2023 Oct 25]. Available from: https://arxiv.org/abs/1506.02640.
31. Sarda A, Dixit S, Bhan A. Object detection for autonomous driving using YOLO algorithm [Internet]. 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM). London, United Kingdom: IEEE; 2021 [cited 2023 Dec 8]:447–51. Available from: https://ieeexplore.ieee.org/document/9445365/.
32. Van Rossum G, Drake FL Jr. Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam; 1995.
33. Chollet F. Keras [Internet]. GitHub, Inc.; 2015 [cited 2023 Dec 8]. Available from: https://github.com/fchollet/keras.
34. Lowekamp BC, Chen DT, Ibáñez L, et al. The design of SimpleITK. Front neuroinform [Internet] 2013;7 [cited 2021 Nov 18]. Available from: http://journal.frontiersin.org/article/10.3389/fninf.2013.00045/abstract.
35. Brett M, Markiewicz CJ, Hanke M, et al. nipy/nibabel: 5.1.0 [Internet]. Geneva (Switzerland): CERN; 2023 [cited 2023 Nov 29]. Available from: https://zenodo.org/record/7795644.
36. Reis D, Kupec J, Hong J, et al. Real-time flying object detection with YOLOv8. arXiv:2305.09972v1 [Preprint]. 2023 [cited 2023 Oct 24]. Available from: https://arxiv.org/abs/2305.09972.
37. Zong Z, Song G, Liu Y. DETRs with collaborative hybrid assignments training. arXiv:2211.12860v6 [Preprint]. 2022 [cited 2023 Oct 27]. Available from: https://arxiv.org/abs/2211.12860.
38. Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med Imaging 2015;15:29.
39. Dice LR. Measures of the amount of ecologic association between species. Ecology 1945;26:297-302.
40. Jaccard P. The distribution of the flora in the alpine zone.1. New Phytol 1912;11:37-50.
41. Ralescu A. Probability and fuzziness. Inf Sci 1984;34:85-92.
42. Minaee S, Boykov YY, Porikli F, et al. Image segmentation using deep learning: a survey. IEEE Trans Pattern Anal Mach Intell 2022;44:3523-42.
43. Csurka G, Volpi R, Chidlovskii B. Semantic image segmentation: two decades of research. arXiv:2302.06378v1 [Preprint]. 2023 [cited 2023 Oct 27]. Available from: https://arxiv.org/abs/2302.06378.
44. Cardenas CE, Yang J, Anderson BM, et al. Advances in auto-segmentation. Semin Radiat Oncol 2019;29:185-97.
45. Sara Mahdavi S, Chng N, Spadinger I, et al. Semi-automatic segmentation for prostate interventions. Med Image Anal 2011;15:226-37.
46. Shahzad R, Pennig L, Goertz L, et al. Fully automated detection and segmentation of intracranial aneurysms in subarachnoid hemorrhage on CTA using deep learning. Sci Rep 2020;10:21799.
47. Zanier O, Da Mutten R, Vieli M, et al. DeepEOR: automated perioperative volumetric assessment of variable grade gliomas using deep learning. Acta Neurochir 2022;165:555-66.
48. Da Mutten R, Zanier O, Ciobanu-Caraus O, et al. Automated volumetric assessment of pituitary adenoma. Endocrine 2024;83:171-7.
49. Hohn EA, Chu B, Martin A, et al. The pedicles are not the densest regions of the lumbar vertebrae: implications for bone quality assessment and surgical treatment strategy. Global Spine J 2017;7:567-71.
50. Siemionow K, Luciano C, Forsthoefel C, et al. Autonomous image segmentation and identification of anatomical landmarks from lumbar spine intraoperative computed tomography scans using machine learning: a validation study. J Craniovertebr Junction Spine 2020;11:99-103.
51. He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv:1406.4729v4 [Preprint]. 2014 [cited 2023 Nov 30]. Available from: https://arxiv.org/abs/1406.4729.