

Original software publication

landmarker: A Toolkit for Anatomical Landmark Localization in 2D/3D Images

Jef Jonkers^{a,*,}, Luc Duchateau^b, Glenn Van Wallendael^c, Sofie Van Hoecke^c^a IDLab, Department of Electronics and Information Systems, Ghent University, Belgium^b Biometrics Research Group, Department of Morphology, Imaging, Orthopedics, Rehabilitation and Nutrition, Ghent University, Belgium^c IDLab, Department of Electronics and Information Systems, Ghent University - imec, Belgium

ARTICLE INFO

Keywords:

Landmark localization
Keypoint detection
Medical image analysis
Python
PyTorch

ABSTRACT

Anatomical landmark localization in 2D/3D images is a critical task in medical imaging. Although many general-purpose tools exist for landmark localization in classical computer vision tasks, such as pose estimation, they lack the specialized features and modularity necessary for anatomical landmark localization applications in the medical domain. Therefore, we introduce **landmarker**, a Python package built on PyTorch. The package provides a comprehensive, flexible toolkit for developing and evaluating landmark localization algorithms, supporting a range of methodologies, including static and adaptive heatmap regression. **landmarker** enhances the accuracy of landmark identification, streamlines research and development processes, and supports various image formats and preprocessing pipelines. Its modular design allows users to customize and extend the toolkit for specific datasets and applications, accelerating innovation in medical imaging. **landmarker** addresses a critical need for precision and customization in landmark localization tasks not adequately met by existing general-purpose pose estimation tools.

Code metadata

Current code version	v0.2.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-24-00590
Permanent link to Reproducible Capsule	Not available
Legal Code License	MIT
Code versioning system used	git
Software code languages, tools, and services used	Python3.10+
Compilation requirements, operating environments & dependencies	PDM is used for dependency management, package installation, and publishing to PyPi
If available Link to developer documentation/manual	https://predict-idlab.github.io/landmarker/
Support email for questions	jef.jonkers@ugent.be

1. Motivation and significance

Landmark (or keypoint) localization in 2D/3D images is a fundamental challenge in computer vision, crucial for tasks such as pose estimation [1–3], face alignment [4,5], robotic manipulation [6], surgical procedures [7], image registration [8], various image-based medical diagnoses [9–13] and treatment response modeling in longitudinal images [14,15]. The effectiveness of different approaches is often domain and task-dependent, necessitating extensive experimentation for new applications. Popular methods, such as heatmap regression, involve

numerous preprocessing and post-processing steps. The latter, such as sub-pixel accuracy methods, can significantly improve the precision needed in medical applications but nevertheless are frequently overlooked.

As existing packages such as OpenPose [16] and Ultralytics [17] are primarily designed for pose estimation, they lack the modularity needed for medical applications. While MMPose [18] offers different components for various tasks, it is still tailored toward pose estimation

* Corresponding author.

E-mail address: jef.jonkers@ugent.be (J. Jonkers).<https://doi.org/10.1016/j.softx.2025.102165>

Received 4 November 2024; Received in revised form 27 March 2025; Accepted 10 April 2025

Available online 2 May 2025

2352-7110/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Table 1
Comparison of Python packages for keypoint/landmark localization.

Package	Custom	3D		Medical		
		Keypoints	Images	Modular	Format	Aug.
ultralytics [17]	✗	✗	✗	✗	✗	✗
MMPose [18]	✓	✓	✗	✗	✗	✗
landmarker	✓	✓	✓	✓	✓	✓

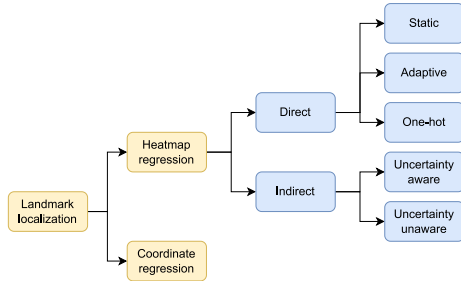


Fig. 1. Taxonomy of (deep learning) landmark localization approaches. The frequently used taxonomy in the literature is highlighted in yellow, while our extended taxonomy of the problem is highlighted in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and does not easily integrate with other frameworks or handle specific medical imaging requirements. Although MMPose [18] supports 3D landmark localization, it cannot process 3D input data, a critical requirement for medical imaging applications. Furthermore, while MMPose offers some flexibility in model creation, its pipeline mandates the use of its framework for preprocessing, training, and evaluation, limiting compatibility with deep learning frameworks such as PyTorch Lightning [19]. Additionally, neither pose estimation library supports medical imaging formats (e.g., NIfTI and DICOM) or provides specialized data augmentation techniques to simulate common acquisition artifacts. Table 1 summarizes the key differences between these existing packages and the proposed landmark detection package within the medical imaging context.

Our Python package, *landmarker*, addresses the need for precise anatomical landmark localization in medical images, which is critical for diagnostics and therapeutic procedures in different specialties, such as orthodontics, maxillofacial surgery, and orthopedics. It is a flexible PyTorch-based toolkit designed explicitly for training, modeling, and evaluating anatomical landmark localization problems. This toolkit accelerates the development of algorithms and enhances the accuracy of landmark identification in medical images, thereby improving diagnostic and treatment outcomes. Its modular and adaptable framework allows researchers and practitioners to implement state-of-the-art algorithms tailored to their specific datasets.

landmarker has a modular framework, which supports various landmark localization algorithms. This flexibility enables customization and extension according to specific needs. The package offers interfaces for data preprocessing, model training, evaluation, and visualization.

2. Background: landmark localization

There are two primary methodologies in landmark localization (see Fig. 1): heatmap regression and coordinate regression. Coordinate regression [20] involves learning a direct relationship between the image and the landmark locations. However, the introduction of heatmap regression [2], which generally yields better performance, has shifted the research focus. Heatmap regression predicts a heatmap that assigns a likelihood or probability to each (downsized) image pixel. Early heatmap regression methods, termed *direct* heatmap regression, assume a parametric distribution, typically a bivariate Gaussian

or Laplacian distribution. Direct heatmap regression can be further classified into static and adaptive approaches. In static heatmap regression, the heatmap distribution parameters are fixed hyperparameters during training. In adaptive heatmap regression, these parameters are adjusted during training, either by treating them as learnable model parameters [12,13,21] or by using scheduling methods that modify the parameters based on specific evaluation metrics [22]. Another variant of static heatmap regression involves training a model on one-hot heatmap images (masks), essentially parameterizing the heatmap as a categorical distribution and transforming the task into a multi-class classification problem where each pixel represents a class [23].

Recent advancements on heatmap regression have seen the rise of fully convolutional neural networks (CNNs) with differentiable decoding operations, such as the soft-argmax operation [1]. These methods, prominent in facial landmark localization, use the heatmap generation layer as an intermediate layer and the decoding operation as the final layer, optimizing the network with a loss function that compares the decoded coordinates to the ground-truth coordinates. Some of these methods account for the ambiguity in landmark ground truth by incorporating uncertainty-aware loss functions, such as the Gaussian log-likelihood loss [5], or other custom loss functions [4,5,24].

In the medical field, landmark localization often utilizes two-stage approaches [25–28]. The first stage identifies landmarks on a low-resolution image, while the second stage refines this localization using a high-resolution patch, which is a region of interest based on the initial predictions. One of the previously described approaches can be applied in both stages. Another technique, also used in medical imaging, involves segmenting or contouring specific anatomical structures and inferring the landmarks from the resulting masks or contours [29,30].

The diversity of approaches in landmark localization, from basic coordinate regression to sophisticated uncertainty-aware heatmap methods, highlights the complexity of this field, particularly in medical imaging applications. While general-purpose tools like OpenPose and MMPose exist, they primarily focus on human pose estimation and lack the specialized features needed for medical imaging applications. These tools often do not support the full range of heatmap generation and decoding methods, uncertainty quantification, or medical image-specific preprocessing pipelines crucial for anatomical landmark localization. Additionally, existing tools typically do not provide the modularity required to implement and experiment with different approaches, such as two-stage refinement methods commonly used in medical applications. These limitations, combined with the need for high precision in medical contexts, underscore the need for a specialized toolkit to handle the unique challenges of anatomical landmark localization while supporting established and emerging methodologies.

3. Software description

landmarker is a Python package that leverages PyTorch [31] deep learning framework and MONAI [32], a PyTorch-based deep learning framework for healthcare imaging for handling medical image files and transformations.

Users can install *landmarker* via [pip](#). After installation, our [documentation](#) and variety of [examples](#) will guide users in applying the toolkit for their specific needs.

3.1. Software architecture and functionalities

The *landmarker* package is structured into several key modules, each addressing specific aspects of the landmark localization pipeline, see Fig. 2.

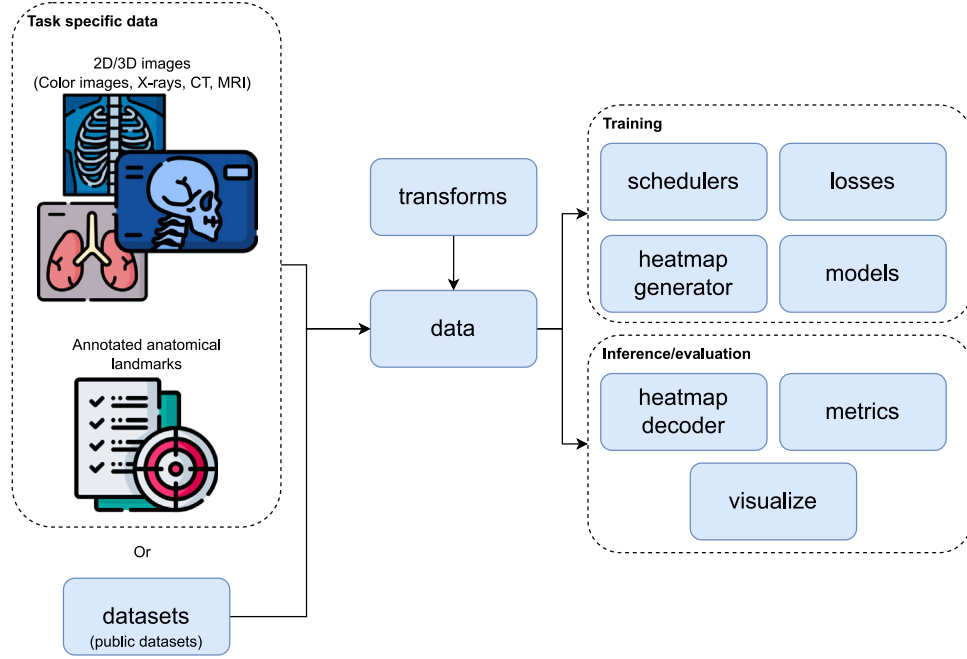


Fig. 2. Flowchart of the different sub-packages of the landmarker package.

3.1.1. Data handling and preprocessing

The data module provides flexible “dataset” classes that inherit from PyTorch’s Dataset, ensuring compatibility with PyTorch DataLoaders. Four main dataset types are supported:

- **LandmarkDataset:** For images with corresponding landmark coordinates.
- **HeatmapDataset:** For images with corresponding static heatmaps representing landmarks.
- **MaskDataset:** For images with binary segmentation masks indicating landmark locations.
- **PatchDataset:** For image patches or regions of interest (ROI) with corresponding landmarks.

These datasets support various image formats (NiftI, DICOM, PNG, JPG, BMP, NPY/NPZ) through integration with the MONAI library [32]. The module can handle single-class, multi-class single-instance, and multi-class multi-instance landmark problems. Utility functions are provided to transform common annotation formats (e.g., labelme [33]) into the required format.

The package also includes the datasets module, which contains functions to import benchmark datasets, such as the ISBI2015 [34] dataset directly. Preprocessing and data augmentation capabilities are available, leveraging MONAI’s transformations for 2D and 3D data. The package handles affine landmarks transformations through the transforms module.

3.1.2. Heatmap generation and decoding

The heatmap module provides functionality for generating target heatmaps and decoding predicted heatmaps, which is essential for heatmap regression approaches.

3.1.2.1. Heatmap generation. The HeatmapGenerator class and its subclasses support various parametric distributions (e.g., multivariate Gaussian and Laplace). The design allows easy implementation of custom distributions. The parameters of the parametric distribution can be set on initialization. Still, they can also be changed during training, for example, for adaptive heatmap regression approaches that rely on a scheduler, such as Adaloss scheduler [22]. Additionally, the parameters could also be learnable parameters during a training procedure.

3.1.2.2. Heatmap decoding. While heatmap generation is only needed for direct approaches, decoding heatmaps is needed for all heatmap regression approaches, with the caveat that the indirect approaches need differentiable decoding operations. Multiple decoding methods are implemented:

The *argmax* operation is the simplest decoder operation [2]. The operation takes the maximum value of the heatmap H_i ,

$$\hat{y}_i = \operatorname{argmax} H_i, \quad (1)$$

to get the coordinate $\hat{y}_i$ of landmark i . The major downside of this approach is that it introduces a discretization error by only choosing a pixel as the predicted landmark.

The *weighted spatial mean* [1,5,24,35] is the approach for indirect heatmap generation. The operation calculates the landmark location $\hat{y}_i$ by first post-processing the heatmap H_i with an activation function σ , typically a type of normalization operation, and afterward taking the weighted spatial mean, e.g., the two-dimensional case:

$$\hat{y}_i = \left(\sum_{c=0}^W \sum_{l=0}^H \frac{c}{W} \sigma(H_i)_{l,c}, \sum_{c=0}^W \sum_{l=0}^H \frac{l}{H} \sigma(H_i)_{l,c} \right) \quad (2)$$

where W and H are, respectively, the heatmap’s width and height. Frequently used activation functions σ are a ReLU activation combined with a normalization procedure and the softmax operation on the heatmap, which is in the literature often referred to as soft-argmax operation [1,35,36].

A significant issue of taking the weighted spatial mean is that it can globally lead to semantically unstructured outputs and thus decrease performance [36]. In [36], they propose to apply soft-argmax operation locally, i.e., a small window, around the heatmap location with a maximum heatmap value. We implement this method and extend it to be used with any post-processing function, i.e., *local weight spatial mean* operation.

3.1.3. Models and losses

While landmarker is compatible with any PyTorch model or loss function, it includes implementations of successful approaches from the literature, such as the spatial configuration network [12], which is effective for positionally consistent medical images like cephalograms.

```

1 from landmarker.data import LandmarkDataset
2
3 image_paths = ... # list of paths of images
4 landmarks = ... # numpy array or torch tensor provided
5 pixel_spacing = ... # pixel spacing of the images
6 class_names = ... # names of the landmarks
7 compose_transform = ... # MONAI compose transform
8
9 ds = LandmarkDataset(
10     imgs=image_paths,
11     landmarks=landmarks,
12     spatial_dims=2,
13     transform=compose_transform,
14     dim_img=(512,512) # dimension to resize to
15     class_names=class_names)

```

Listing 1 Loading data into a LandmarkDataset

```

1 from landmarker.dataset import get_cepha_landmark_datasets
2
3 data_dir = ...
4 ds_train, ds_test1, ds_test2 = get_cepha_landmark_datasets(data_dir)

```

Listing 2 Loading data into a LandmarkDataset through the dataset module.

```

1 from landmarker.heatmap import GaussianHeatmapGenerator
2
3 generator = GaussianHeatmapGenerator(
4     nb_landmarks=19, # number of landmarks
5     sigmas=3, # sd. value of the Gaussian heatmap function
6     learnable=True, # make the covariance matrix learnable parameters
7     heatmap_size=(512,512))

```

Listing 3 Setting up Gaussian heatmap generator.

Also, multiple heatmap regression loss functions are implemented from successful approaches [4,5,13,37–39].

3.1.4. Evaluation and visualization

The visualize module allows inspecting constructed datasets and trained models. Additionally, it enables the generation of detection reports that output several metrics from the metrics module, such as point error and success detection rate (SDR), which is the percentage of predicted landmarks with a point error smaller or equal to a specified radius.

4. Illustrative examples

In this section, we illustrate the use of `landmarker`. We will perform adaptive heatmap regression with the ISBI2015 cephalometric landmark dataset, showcasing most functionality.

We start by loading the data into a `LandmarkDataset` by providing a list of paths to images, a NumPy array of the shape (N, C, D) where N is the number of samples, C is the number of landmark classes, and D represents the spatial dimension which can be 2 or 3, see Listing 1.

Alternatively, one can also use the `dataset` module, which includes everything to import the ISBI2015 dataset directly, see Listing 2.

Once the data is loaded in the proper format, we can set up the heatmap generator to generate a heatmap from landmarks, see Listing 3.

After this, the user can visually inspect if everything is initialized as expected by using the `inspection_plot` function (Listing 4 and Fig. 3).

With the `LandmarkDataset` and `HeatmapGenerator` adequately set up, we can now train a model to output likelihood heatmaps.

In Listing 5, we train a `SpatialConfigurationNetwork` [12,13] and parameterization of the Gaussian heatmap [13,21].

After the training procedure, one can visually evaluate the results by using the `prediction_inspect_plot` function (see Listing 6 and Fig. 4) or by creating a detection report.

4.1. Benchmarks

To evaluate our package capabilities and its model's landmark localization precision quantitatively, we conducted two benchmark experiments on relatively recent datasets that showcase its performance on both 2D and 3D medical imaging tasks. For the 2D benchmark, we used pelvis x-ray images from the Osteoarthritis Initiative [11]. In this experiment, our spatial configuration network (SCN) with adaptive heatmaps—implemented in `landmarker`, was compared against a state-of-the-art U-Net with Attention approach [11]. As shown in Table 2, our method achieved a lower average position error (PE) and higher success detection rates (SDR) across multiple thresholds, clearly demonstrating the capabilities of the package as a benchmarking tool.

In addition to the 2D evaluation, we benchmarked our model on 3D images using CT scans of the skull [40], where the task was to annotate mandibular landmarks automatically. The results, presented in Table 3, include both validation and test metrics, with our one-hot ensemble [41] method outperforming the competitive baseline regarding PE and SDR. These results confirm our model's precision and highlight its flexibility in handling diverse anatomical landmarking applications across different imaging modalities.

5. Impact and conclusions

The development and release of the `landmarker` Python package offer significant advancements in anatomical landmark localization, particularly within medical imaging. By providing a flexible and

```

1 from landmarker.visualize import inspection_plot
2
3 inspection_plot(ds_train, range(3), heatmap_generator=heatmap_generator)

```

Listing 4 Loading data into a LandmarkDataset

```

1 import torch
2
3 from landmarker.models import OriginalSpatialConfigurationNet
4 from landmarker.losses import GaussianHeatmapL2Loss
5
6 model = OriginalSpatialConfigurationNet(in_channels=1, out_channels=19)
7 optimizer = torch.optim.SGD([
8     {'params': model.parameters(), "weight_decay":1e-3},
9     {'params': heatmap_generator.sigmas},
10    {'params': heatmap_generator.rotation}]
11    , lr=1e-6, momentum=0.99, nesterov=True)
12
13 criterion = GaussianHeatmapL2Loss(alpha=5)
14
15 train_loader = DataLoader(ds_train, batch_size=1, shuffle=True, num_workers=0)
16
17 # Start training
18 for epoch in range(100):
19     model.train()
20     for batch in train_loader:
21         images = batch["image"].to(device)
22         landmarks = batch["landmark"].to(device)
23         optimizer.zero_grad()
24         outputs = model(images)
25         heatmaps = heatmap_generator(landmarks)
26         loss = criterion(outputs, heatmap_generator.sigmas, heatmaps)
27         loss.backward()
28         optimizer.step()

```

Listing 5 Training adaptive heatmap regression model.

```

1 from landmarker.visualize import prediction_inspect_plot
2
3 prediction_inspect_plot(ds_test1, model, ds_test1.indices[:3])

```

Listing 6 Prediction inspection plot.

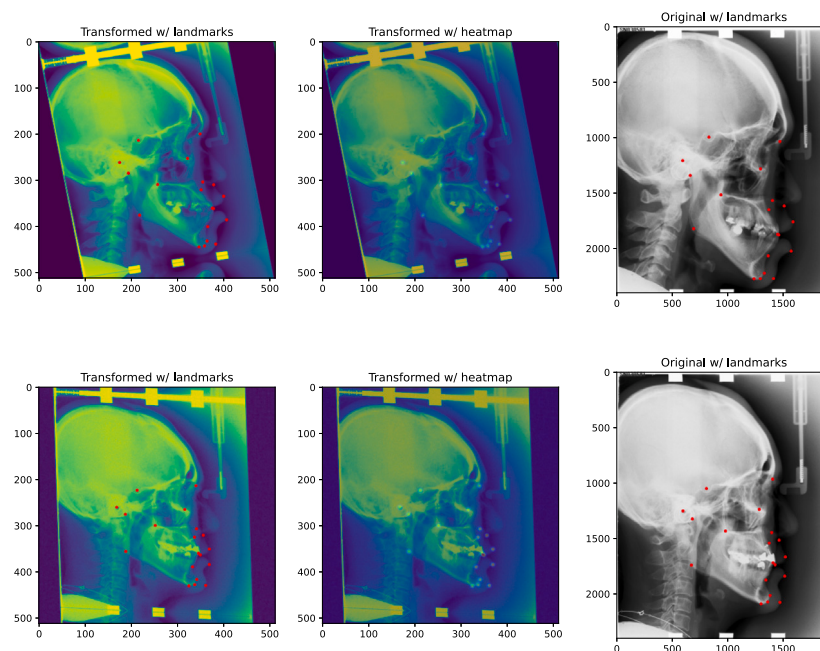


Fig. 3. Example of results of running the inspection_plot on the ISBI2015 dataset.

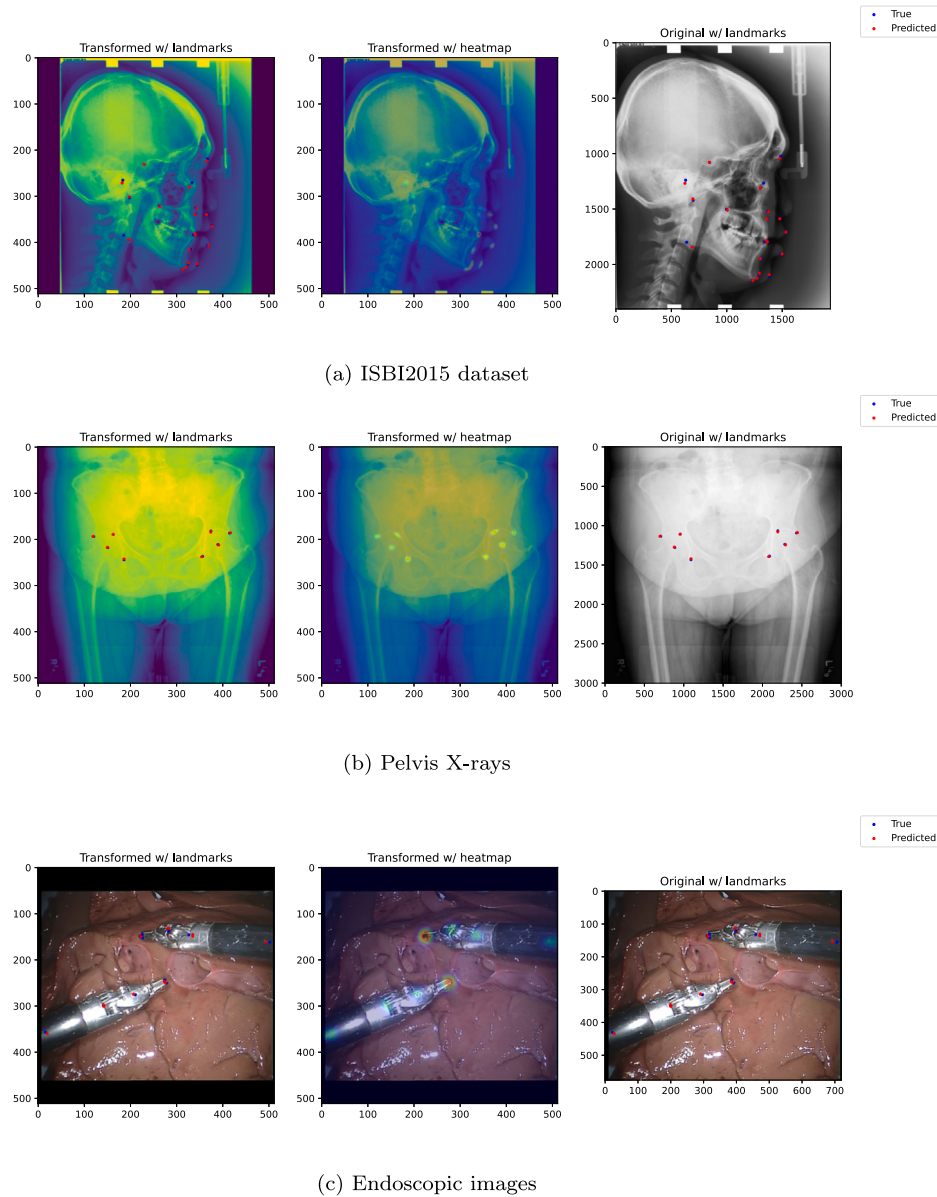


Fig. 4. Examples of landmark predictions using the `prediction_inspection_plot` tool for different datasets: (a) Endoscopic images, (b) ISBI2015, and (c) Pelvis X-rays.

Table 2

Landmark localization results of spatial configuration network (SCN) with adaptive heatmaps [13,21] (implemented in `landmarker`) on the pelvis x-rays (2D) of the osteoarthritis initiative dataset [11] compared against the literature. The best results are highlighted in bold. A 5-fold validation approach, as suggested by Pei et al. [11], obtains the results.

Model name	PE (mm)	SDR (%)			
		1 mm	2 mm	3 mm	4 mm
U-Net w/ Attention [11]	3.14	8.65%	33.89%	59.25%	78.61%
SCN [13,21] (<code>landmarker</code>)	1.61	33.25%	74.38%	90.96%	95.85%

modular toolkit tailored explicitly for medical imaging applications, `landmarker` addresses a critical need for precision and customization in landmark localization tasks that are not adequately met by existing general-purpose pose estimation tools like OpenPose or MMPose.

One of the key impacts of `landmarker` is its ability to streamline the research and development process for researchers and developers working on image-based diagnostic and therapeutic deep learning applications. The package's integration with PyTorch and MONAI ensures compatibility with various medical image formats and processing pipelines, facilitating easy adoption in research environments. Moreover, the extensive support for different landmark localization methodologies, including static and adaptive heatmap regression, allows users to implement state-of-the-art algorithms with minimal overhead.

Including customizable modules for data handling, preprocessing, heatmap generation, and model evaluation further enhances the utility of `landmarker`. These features enable users to quickly adapt the package to their specific use cases, whether they involve 2D or 3D images, single-class or multi-class landmark localization, or static versus adaptive methodologies. The ability to implement and test novel approaches within a unified framework accelerates innovation in computer-aided diagnosis through medical imaging, potentially leading to more accurate diagnoses and better patient outcomes.

Table 3

Landmark localization results of one hot ensemble [41] (implemented in *landmarker*) on the MML (3D) dataset compared against the literature. This benchmark dataset is the same as the data subset, only with complete landmarks, used as a benchmark in He et al. [40]. The best results are highlighted in bold.

Model name	Validation					Test				
	PE (mm)	SDR (%)				PE (mm)	SDR (%)			
		2 mm	2.5 mm	3 mm	4 mm		2 mm	2.5 mm	3 mm	4 mm
Pruning-ResUNet3D [40]	1.82	73.21%	82.14%	88.93%	94.76%	1.96	70.03%	79.97%	86.10%	92.73%
One hot ensemble [41] (landmarker)	1.60	77.81%	87.76%	89.92%	93.37%	1.39	81.67%	91.31%	93.33%	96.31%

In conclusion, *landmarker* represents a powerful tool for advancing the field of anatomical landmark localization in medical imaging. Its flexibility and ease of use make it a helpful resource for researchers and engineers. We plan to incorporate future developments and improvements, particularly in uncertainty quantification, which is critical for enhancing the reliability and robustness of landmark localization in medical applications.

CRedit authorship contribution statement

Jef Jonkers: Writing – original draft, Software, Methodology, Conceptualization. **Luc Duchateau:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Glenn Van Wallendael:** Writing – review & editing, Supervision, Conceptualization. **Sofie Van Hoecke:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Funding sources

Jef Jonkers is funded by the Research Foundation Flanders (FWO, Ref. 1S11525N). This research was also (partially) funded by the Flemish Government (AI Research Program).

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jef Jonkers reports financial support was provided by Research Foundation Flanders (FWO). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Luvizon DC, Picard D, Tabia H. 2D/3D pose estimation and action recognition using multitask deep learning. 2018, <http://dx.doi.org/10.48550/arXiv.1802.09232>, arXiv:1802.09232 [cs]. URL <http://arxiv.org/abs/1802.09232>.
- [2] Tompson JJ, Jain A, LeCun Y, Bregler C. Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in neural information processing systems. vol. 27, Curran Associates, Inc.; 2014, URL <https://proceedings.neurips.cc/paper/2014/hash/e744f91c29ec99f0e662c9177946c627-Abstract.html>.
- [3] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer vision – ECCV 2016. Lecture notes in computer science, Cham: Springer International Publishing; 2016, p. 483–99. http://dx.doi.org/10.1007/978-3-319-46484-8_29.
- [4] Zhou Z, Li H, Liu H, Wang N, Yu G, Ji R. STAR loss: Reducing semantic ambiguity in facial landmark detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 15475–84, URL https://openaccess.thecvf.com/content/CVPR2023/html/Zhou_STAR_Loss_Reducing_Semantic_Ambiguity_in_Facial_Landmark_Detection_CVPR_2023_paper.html.
- [5] Kumar A, Marks TK, Mou W, Feng C, Liu X. UGLLI face alignment: Estimating uncertainty with Gaussian log-likelihood loss. In: 2019 IEEE/CVF international conference on computer vision workshop (ICCVW). Seoul, Korea (South): IEEE; 2019, p. 778–82. <http://dx.doi.org/10.1109/ICCVW.2019.00103>, URL <https://ieeexplore.ieee.org/document/9022191/>.
- [6] Ziegler T, Butepage J, Welle MC, Varava A, Novkovic T, Kragic D. Fashion landmark detection and category classification for robotics. 2020, <http://dx.doi.org/10.48550/arXiv.2003.11827>, arXiv:2003.11827 [cs, stat]. URL <http://arxiv.org/abs/2003.11827>.
- [7] Sharan L, Romano G, Brand J, Kelm H, Karck M, De Simone R, et al. Point detection through multi-instance deep heatmap regression for sutures in endoscopy. Int J Comput Assist Radiol Surg 2021;16(12):2107–17. <http://dx.doi.org/10.1007/s11548-021-02523-w>.
- [8] Grewal M, Wiersma J, Westerveld H, Bosman PAN, Alderliesten T. Automatic landmark correspondence detection in medical images with an application to deformable image registration. J Med Imaging 2023;10(1):014007. <http://dx.doi.org/10.1117/1.JMI.10.1.014007>, Publisher: SPIE. URL <https://www.spiedigitallibrary.org/journals/journal-of-medical-imaging/volume-10/issue-1/014007/Automatic-landmark-correspondence-detection-in-medical-images-with-an-application/10.1117/1.JMI.10.1.014007.full>.
- [9] de Queiroz Tavares Borges Mesquita G, Vieira WA, Vidigal MTC, Travencolo BAN, Beaini TL, Spin-Neto R, et al. Artificial intelligence for detecting cephalometric landmarks: A systematic review and meta-analysis. J Digit Imaging 2023. <http://dx.doi.org/10.1007/s10278-022-00766-w>.
- [10] Ryu SM, Shin K, Shin SW, Lee SH, Seo SM, Cheon S-u, et al. Automated landmark identification for diagnosis of the deformity using a cascade convolutional neural network (FlatNet) on weight-bearing lateral radiographs of the foot. Comput Biol Med 2022;148:105914. <http://dx.doi.org/10.1016/j.combiomed.2022.105914>, URL <https://www.sciencedirect.com/science/article/pii/S0010482522006576>.
- [11] Pei Y, Mu L, Xu C, Li Q, Sen G, Sun B, et al. Learning-based landmark detection in pelvis X-rays with attention mechanism: data from the osteoarthritis initiative. Biomed Phys Eng Express 2023;9(2):025001. <http://dx.doi.org/10.1088/2057-1976/ac8ffa>.
- [12] Payer C, Stern D, Bischof H, Urschler M. Integrating spatial configuration into heatmap regression based CNNs for landmark localization. Med Image Anal 2019;54:207–19. <http://dx.doi.org/10.1016/j.media.2019.03.007>, URL <https://www.sciencedirect.com/science/article/pii/S1361841518305784>.
- [13] Thaler F, Payer C, Urschler M, Stern D. Modeling annotation uncertainty with Gaussian heatmaps in landmark localization. 2021, <http://dx.doi.org/10.48550/arXiv.2109.09533>, arXiv:2109.09533 [cs] URL <http://arxiv.org/abs/2109.09533>.
- [14] Ou Y, Weinstein SP, Conant EF, Englander S, Da X, Gaonkar B, et al. Deformable registration for quantifying longitudinal tumor changes during neoadjuvant chemotherapy. Magn Reson Med 2015;73(6):2343–56. <http://dx.doi.org/10.1002/mrm.25368>, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.25368>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.25368>.
- [15] Cai J, Tang Y, Yan K, Harrison AP, Xiao J, Lin G, et al. Deep lesion tracker: Monitoring lesions in 4D longitudinal imaging studies. 2021, p. 15159–69, URL https://openaccess.thecvf.com/content/CVPR2021/html/Cai_Deep_Lesion_Tracker_Monitoring_Lesions_in_4D_Longitudinal_Imaging_Studies_CVPR_2021_paper.html.
- [16] Cao Z, Hidalgo G, Simon T, Wei S-E, Sheikh Y. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity fields. IEEE Trans Pattern Anal Mach Intell 2021;43(1):172–86. <http://dx.doi.org/10.1109/TPAMI.2019.2929257>, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. URL <https://ieeexplore.ieee.org/document/8765346>.
- [17] Jocher G, Qiu J, Chaurasia A. Ultralytics YOLO. 2023, original-date: 2022-09-11T16:39:45Z. URL <https://github.com/ultralytics/ultralytics>.
- [18] MMPose Contributors. OpenMMLab pose estimation toolbox and benchmark. 2020, original-date: 2020-07-08T06:02:55Z. URL <https://github.com/open-mmlab/mmpose>.
- [19] Falcon W, The PyTorch Lightning team. PyTorch Lightning. 2019, <http://dx.doi.org/10.5281/zenodo.3828935>, URL <https://github.com/Lightning-AI/lightning>.
- [20] Fard AP, Mahoor MH. ACR loss: Adaptive coordinate-based regression loss for face alignment. In: 2022 26th international conference on pattern recognition (ICPR). 2022, p. 1807–14. <http://dx.doi.org/10.1109/ICPR56361.2022.9956683>.
- [21] Payer C, Urschler M, Bischof H, Stern D. Uncertainty estimation in landmark localization based on Gaussian heatmaps. In: Sudre CH, Fehri H, Arbel T, Baumgartner CF, Dalca A, Tanno R, Van Leemput K, Wells WM, Sotiras A, Papiez B, Ferrante E, Parisot S, editors. Uncertainty for safe utilization of machine learning in medical imaging, and graphs in biomedical image analysis.

- Lecture notes in computer science, Cham: Springer International Publishing; 2020, p. 42–51. http://dx.doi.org/10.1007/978-3-030-60365-6_5.
- [22] Teixeira B, Tamersoy B, Singh V, Kapoor A. Adaloss: Adaptive Loss Function for Landmark Localization. 2019, <http://dx.doi.org/10.48550/arXiv.1908.01070>, arXiv:1908.01070 [cs]. URL <http://arxiv.org/abs/1908.01070>.
- [23] McCouat J, Voiculescu I. Contour-hugging heatmaps for landmark detection. In: 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). 2022, p. 20565–73. <http://dx.doi.org/10.1109/CVPR52688.2022.01994>, URL <https://ieeexplore.ieee.org/document/9879107>.
- [24] Kumar A, Marks TK, Mou W, Wang Y, Jones M, Cherian A, et al. LUVLi face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 8236–46, URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kumar_LUVLi_Face_Alignment_Estimating_Landmarks_Location_Uncertainty_and_Visibility_Likelihood_CVPR_2020_paper.html.
- [25] Jiang Y, Li Y, Wang X, Tao Y, Lin J, Lin H. CephalFormer: Incorporating global structure constraint into visual features for general cephalometric landmark Detection. In: Wang L, Dou Q, Fletcher PT, Speidel S, Li S, editors. Medical image computing and computer assisted intervention – MICCAI 2022. Lecture notes in computer science, Cham: Springer Nature Switzerland; 2022, p. 227–37. http://dx.doi.org/10.1007/978-3-031-16437-8_22.
- [26] Song Y, Qiao X, Iwamoto Y, Chen Y-w. Automatic Cephalometric landmark detection on X-ray images using a deep-learning method. Appl Sci 2020;10(7):2547. <http://dx.doi.org/10.3390/app10072547>, Number: 7 Publisher: Multidisciplinary Digital Publishing Institute. URL <https://www.mdpi.com/2076-3417/10/7/2547>.
- [27] Zhong Z, Li J, Zhang Z, Jiao Z, Gao X. An attention-guided deep regression model for landmark detection in Cephalograms. In: Shen D, Liu T, Peters TM, Staib LH, Essert C, Zhou S, Yap P-T, Khan A, editors. Medical image computing and computer assisted intervention – MICCAI 2019. Lecture notes in computer science, Cham: Springer International Publishing; 2019, p. 540–8. http://dx.doi.org/10.1007/978-3-030-32226-7_60.
- [28] Nguyen TP, Chae D-S, Park S-J, Kang K-Y, Lee W-S, Yoon J. Intelligent analysis of coronal alignment in lower limbs based on radiographic image with convolutional neural network. Comput Biol Med 2020;120:103732. <http://dx.doi.org/10.1016/j.combiomed.2020.103732>, URL <https://www.sciencedirect.com/science/article/pii/S0010482520301153>.
- [29] Gooßen A, Hermann E, Weber GM, Gernoth T, Pralow T, Grigat R-R. Model-based segmentation of pediatric and adult joints for orthopedic measurements in digital radiographs of the lower limbs. Comput Sci - Res Dev 2011;26(1):107–16. <http://dx.doi.org/10.1007/s00450-010-0139-8>.
- [30] Pei Y, Yang W, Wei S, Cai R, Li J, Guo S, et al. Automated measurement of hip-knee-ankle angle on the unilateral lower limb X-rays using deep learning. Phys Eng Sci Med 2021;44(1):53–62. <http://dx.doi.org/10.1007/s13246-020-00951-7>.
- [31] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. 2019, arXiv:1912.01703 [cs, stat]. URL <http://arxiv.org/abs/1912.01703>.
- [32] Cardoso MJ, Li W, Brown R, Ma N, Kerfoot E, Wang Y, et al. MONAI: An open-source framework for deep learning in healthcare. 2022, <http://dx.doi.org/10.48550/arXiv.2211.02701>, arXiv:2211.02701 [cs]. URL <http://arxiv.org/abs/2211.02701>.
- [33] Wada K. Labelme: Image polygonal annotation with Python. 2024, <http://dx.doi.org/10.5281/zenodo.5711226>, URL <https://github.com/wkentaro/labelme>.
- [34] Wang C-W, Huang C-T, Lee J-H, Li C-H, Chang S-W, Siao M-J, et al. A benchmark for comparison of dental radiography analysis algorithms. Med Image Anal 2016;31:63–76. <http://dx.doi.org/10.1016/j.media.2016.02.004>, URL <https://www.sciencedirect.com/science/article/pii/S1361841516000190>.
- [35] Dong X, Yu S-I, Weng X, Wei S-E, Yang Y, Sheikh Y. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. 2018, p. 360–8, URL https://openaccess.thecvf.com/content_cvpr_2018/html/Dong_Supervision-by-Registration_An_Unsupervised_CVPR_2018_paper.html.
- [36] Bulat A, Sanchez E, Tzimiropoulos G. Subpixel heatmap regression for facial landmark localization. 2021, <http://dx.doi.org/10.48550/arXiv.2111.02360>, arXiv:2111.02360 [cs]. URL <http://arxiv.org/abs/2111.02360>.
- [37] Nibali A, He Z, Morgan S, Prendergast L. Numerical coordinate regression with convolutional neural networks. 2018, arXiv:1801.07372 [cs]. URL <http://arxiv.org/abs/1801.07372>.
- [38] Feng Z-H, Kittler J, Awais M, Huber P, Wu X-J. Wing loss for robust facial landmark localisation with convolutional neural networks. 2018, p. 2235–45, URL https://openaccess.thecvf.com/content_cvpr_2018/html/Feng_Wing_Loss_for_CVPR_2018_paper.html.
- [39] Wang X, Bo L, Fuxin L. Adaptive wing loss for robust face alignment via heatmap regression. 2019, p. 6971–81, URL https://openaccess.thecvf.com/content_ICCV_2019/html/Wang_Adaptive_Wing_Loss_for_Robust_Face_Alignment_via_Heatmap_Regression_ICCV_2019_paper.html.
- [40] He T, Xu G, Cui L, Tang W, Long J, Guo J. Anchor ball regression model for large-scale 3D skull landmark detection. Neurocomputing 2024;567:127051. <http://dx.doi.org/10.1016/j.neucom.2023.127051>, URL <https://www.sciencedirect.com/science/article/pii/S0925231223011748>.
- [41] Jonkers J, Coopman F, Duchateau L, Wallendaal GV, Hoecke SV. Reliable uncertainty quantification for 2D/3D anatomical landmark localization using multi-output conformal prediction. 2025, <http://dx.doi.org/10.48550/arXiv.2503.14106>, arXiv:2503.14106 [cs]. URL <http://arxiv.org/abs/2503.14106>.