



# Joint Dense-Point Representation for Contour-Aware Graph Segmentation

Kit Mills Bransby<sup>1</sup>(✉), Greg Slabaugh<sup>1</sup>, Christos Bourantas<sup>1,2</sup>,  
and Qianni Zhang<sup>1</sup>

<sup>1</sup> Queen Mary University of London, London, UK

{k.m.bransby, qianni.zhang}@qmul.ac.uk

<sup>2</sup> Department of Cardiology, Barts Health NHS Trust, London, UK

**Abstract.** We present a novel methodology that combines graph and dense segmentation techniques by jointly learning both point and pixel contour representations, thereby leveraging the benefits of each approach. This addresses deficiencies in typical graph segmentation methods where misaligned objectives restrict the network from learning discriminative vertex and contour features. Our joint learning strategy allows for rich and diverse semantic features to be encoded, while alleviating common contour stability issues in dense-based approaches, where pixel-level objectives can lead to anatomically implausible topologies. In addition, we identify scenarios where correct predictions that fall on the contour boundary are penalised and address this with a novel hybrid contour distance loss. Our approach is validated on several Chest X-ray datasets, demonstrating clear improvements in segmentation stability and accuracy against a variety of dense- and point-based methods. Our source code is freely available at:

[www.github.com/kitbransby/Joint\\_Graph\\_Segmentation](https://www.github.com/kitbransby/Joint_Graph_Segmentation).

**Keywords:** Semantic Segmentation · Graph Convolutional Networks

## 1 Introduction

Semantic segmentation is a fundamental task in medical imaging used to delineate regions of interest, and has been applied extensively in diagnostic radiology. Recently, deep learning methods that use a dense probability map to classify each pixel such as UNet [2], R-CNN [3], FCN [4] have advanced the state-of-the-art in this area. Despite overall excellent performance, dense-based approaches learn using a loss defined at the pixel-level which can lead to implausible segmentation boundaries such as unexpected interior holes or disconnected blobs [1]. This is a particular problem in medical image analysis where information-poor, occluded or artefact-affected areas are common and often limit a network's ability to predict reasonable boundaries. Furthermore, minimising the largest error (Hausdorff distance (HD)) is often prioritised over general segmentation metrics such as Dice Similarity (DS) or Jaccard Coefficient (JC) in medical imaging, as stable and trustworthy predictions are more desirable.

To address this problem in segmentation networks, Gaggion *et al.* proposed HybridGNet [1] that replaces the convolutional decoder in UNet with a graph convolutional network (GCN), where images are segmented using a polygon generated from learned points. Due to the relational inductive bias of graph networks where features are shared between neighbouring nodes in the decoder, there is a natural smoothing effect in predictions leading to stable segmentation and vastly reduced HD. In addition this approach is robust to domain shift and can make reasonable predictions on unseen datasets sourced from different medical centres, whereas dense-based methods fail due to domain memorization [5]. In HybridGNet, improved stability and HD comes at the cost of reduced contour detail conveyed by sub-optimal DS and JC metrics when compared to dense-based approaches such as UNet. Many methods have addressed this problem by rasterizing polygon points predicted by a decoder to a dense mask and then training the network using typical pixel-level losses such as Dice or cross-entropy [7,9,10]. These approaches have merit but are often limited by their computational requirements. For example, in CurveGCN [7], the rasterization process uses OpenGL polygon triangulation which is not differentiable, and the gradients need to be approximated using Taylor expansion which is computationally expensive and can therefore only be applied at the fine-tuning stage [8]. While in ACDRNet [10], rasterization is differentiable, however the triangulation process is applicable only to convex polygons, and therefore limits application to more complicated polygon shapes. Rasterization is extended to non-convex polygons in BoundaryFormer [9] by bypassing the triangulation step and instead approximating the unsigned distance field. This method gives excellent results on MS-COCO dataset [11], however is computationally expensive (see Sect. 3.3).

With this in mind, we return to HybridGNet which efficiently optimises points directly and theorise about the causes of the performance gap relative to dense segmentation models. We identify that describing segmentation contours using points is a sub-optimal approach because (1) points are an incomplete representation of the segmentation map; (2) the supervisory signal is usually weaker ( $n$  distances are calculated from  $n$  pairs of points, versus,  $h \times w$  distances for pairs of dense probability maps); (3) the distance from the contour is more meaningful than the distance from the points representing the contour, hence minimising the point-wise distance can lead to predictions which fall on the contour being penalised.

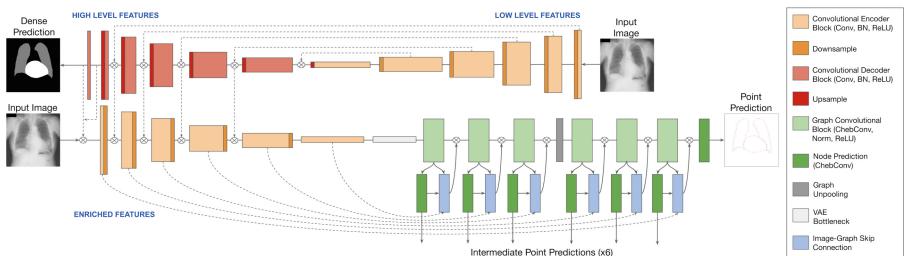
*Contributions:* We propose a novel joint architecture and contour loss to address this problem that leverages the benefits of both point and dense approaches. First, we combine image features from an encoder trained using a point-wise distance with image features from a decoder trained using a pixel-level objective. Our motivation is that contrasting training strategies enable diverse image features to be encoded which are highly detailed, discriminative and semantically rich when combined. Our joint learning strategy benefits from the segmentation accuracy of dense-based approaches, but without topological errors that regularly afflict models trained using a pixel-level loss. Second, we propose a novel hybrid contour distance (HCD) loss which biases the distance field towards pre-

dictions that fall on the contour boundary using a sampled unsigned distance function which is fully differentiable and computationally efficient. To our knowledge this is the first time unsigned distance fields have been applied to graph segmentation tasks in this way. Our approach is able to generate highly plausible and accurate contour predictions with lower HD and higher DS/JC scores than a variety of dense and graph-based segmentation baselines.

## 2 Methods

### 2.1 Network Design

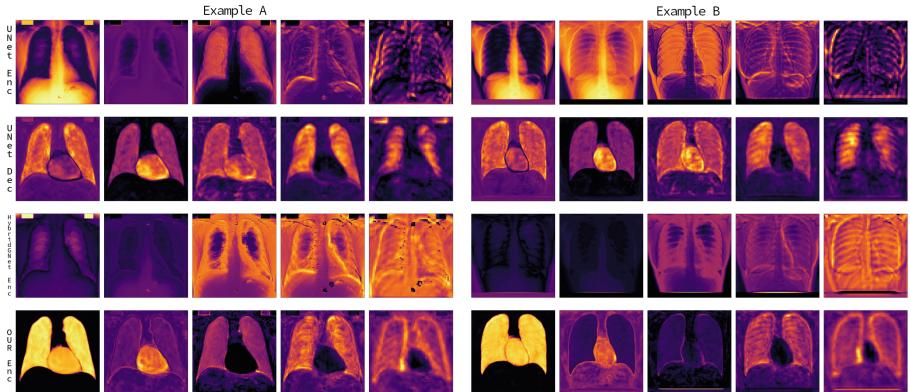
We implement an architecture consisting of two networks, a Dense-Graph (DG) network and a Dense-Dense (DD) network, as shown in Fig. 1. Each network takes the same image input  $X$  of height  $H$  and width  $W$  with skip connections passing information from the decoder of DD to the encoder of DG. For DG, we use a HybridGNet-style architecture containing a convolutional encoder to learn image features at multiple resolutions, and a graph convolutional decoder to regress the 2D coordinates of each point. In DG, node features are initialised in a variational autoencoder (VAE) bottleneck where the final convolutional output is flattened to a low dimensional latent space vector  $z$ . We sample  $z$  from a distribution  $\text{Normal}(\mu, \sigma)$  using the reparameterization trick [12], where  $\mu$  and  $\sigma$  are learnt parameters of the encoder. Image-to-Graph Skip Connections (IGSC) [1] are used to sample dense feature maps  $F_I \in \mathbb{R}^{H \times W \times C}$  from DG’s encoder using node position predictions  $P \in \mathbb{R}^{N \times 2}$  from DG’s graph decoder and concatenate these with previous node features  $F_G \in \mathbb{R}^{N \times f}$  to give new node features  $F'_G \in \mathbb{R}^{N \times (f+C+2)}$ . Here,  $N$  is the number of nodes in the graph and  $f$  is the dimension of the node embedding. We implement IGSC at every encoder-decoder level and pass node predictions as output, resulting in seven node predictions. For DD, we use a standard UNet using the same number of layers and dimensions as the DG encoder with a dense segmentation prediction at the final decoder layer.



**Fig. 1.** Network Architecture: a Dense-Dense network (top) enriches image features in a Dense-Graph network (bottom).

## 2.2 Graph Convolutional Network

Our graph decoder passes features initialised from the VAE bottleneck through six Chebyshev spectral graph convolutional [13] (ChebConv) layers using K-order polynomial filters. Briefly, this is defined by  $X' = \sigma(\sum_{K=1}^k Z^{(k)} \cdot \Theta^{(k)})$  where  $\Theta^{(k)} \in \mathbb{R}^{f_{in} \times f_{out}}$  are learnable weights and  $\sigma$  is a ReLU activation function.  $Z^{(k)}$  is computed recursively such that  $Z^{(1)} = X$ ,  $Z^{(2)} = \hat{L} \cdot Z^{(1)}$ ,  $Z^{(k)} = 2 \cdot \hat{L} \cdot Z^{(k-1)} - Z^{(k-2)}$  where  $X \in \mathbb{R}^{N \times f_{in}}$  are graph features, and  $\hat{L}$  represents the scaled and normalized graph Laplacian [14]. In practice, this allows for node features to be aggregated within a K-hop neighbourhood, eventually regressing the 2D location of each node using additional ChebConv prediction layers ( $f_{out} = 2$ ). As in [1], our graph network also includes an unpooling layer after ChebConv block 3 to upsample the number of points by adding a new point in between existing ones.

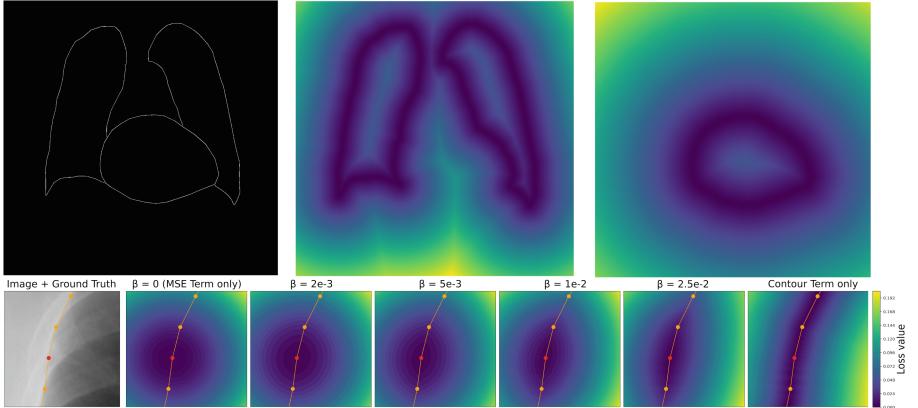


**Fig. 2.** Feature map activation comparison between UNet encoder, UNet decoder, HybridGNet encoder and our encoder, using two examples. Top four most activated channels are summed channel-wise for convolutional layers 1–5 in each encoder/decoder. L→R: decreasing resolution, increasing channel depth. Note, activations in our encoder consistently highlight areas which are more pertinent to segmentation

## 2.3 Joint Dense-Point Learning

As typical DG networks are trained with a point-wise distance loss and not a pixel-level loss, the image encoder is not directly optimised to learn clear and well-defined boundary features. This misalignment problem results in the DG encoder learning features pertinent to segmentation which are distinctively different from those learnt in DD encoders. This is characterised by activation peaks in different image regions such as the background and other non-boundary areas (see Fig. 2). To leverage this observation, we enrich the DG encoder feature maps at multiple scales by fusing them with image features learnt by a DD decoder using a pixel-level loss. These diverse and highly discriminative features

are concatenated before being passed through the convolutional block at each level. Current GCN feature learning paradigms aim at combining feature maps from neighbouring or adjacent levels so as to aggregate similar information. This results in a “coarse-to-fine” approach by first passing high level features to early graph decoder blocks, followed by low level features to late graph decoder blocks. Our joint learning approach is similar to this strategy but also supplements each DG encoder level with both semantically rich and highly detailed contour features learnt by the DD network.



**Fig. 3.** Our Hybrid Contour Distance loss biases the distance field to contours rather than the points representing the contour. Top L–R: Segmentation mask represented with edges, unsigned distance field for lungs, and heart. Bottom: Effect of beta in HCD.

## 2.4 Hybrid Contour Distance

Mean squared error (MSE) is a spatially symmetric loss which is agnostic to true contour borders. We alleviate this pitfall by designing an additional contour-aware loss term that is sensitive to the border. To achieve this we precompute a 2D unsigned distance map  $S$  from the dense segmentation map for each class  $c$  (i.e. lungs, heart), where each position represents the normalised distance to the closest contour border of that class. Specifically, for a dense segmentation map  $M$  we use a Canny filter [15] to find the contour boundary  $\delta M$  and then determine the minimum distance between a point  $x \in c$  and any point  $p$  on the boundary  $\delta M_c$ . This function is positive for both the interior and exterior regions, and zero on the boundary. Our method is visualised in Fig. 3 (first row) and formalised below:

$$S_c(\mathbf{x}) = \min |\mathbf{x} - \mathbf{p}| \text{ for all } \mathbf{p} \in \delta M_c \quad (1)$$

During training, we sample  $S_c$  as an additional supervisory signal using the predicted 2D point coordinates  $\hat{y}_i \in c$ , and combine with MSE with weight  $\beta$ .

The effect of  $\beta$  is illustrated in Fig. 3 (second row) and full HCD loss function is defined below, where  $N$  is the number of points and  $y_i \in c$  is the ground truth point coordinate.

$$\mathcal{L}_{HCD} = \frac{1}{N} \sum_{i=1}^N [(y_i - \hat{y}_i)^2 + \beta S_c(\hat{y}_i)] \quad (2)$$

### 3 Experiments and Results

#### 3.1 Datasets

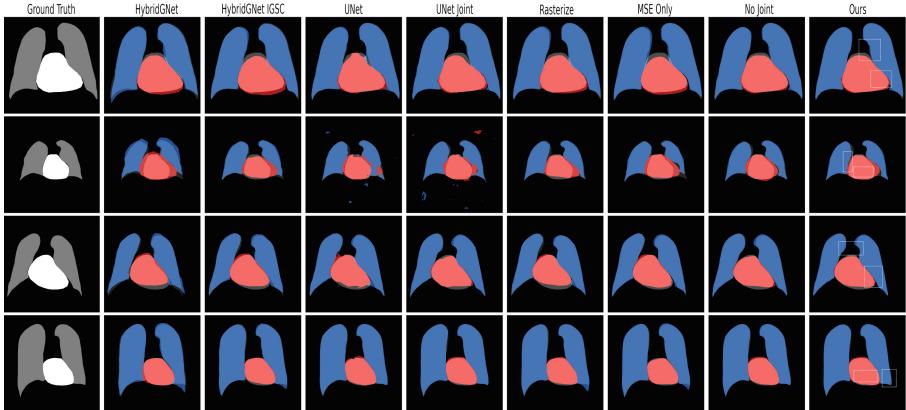
We obtain four publicly available Chest X-ray segmentation datasets (JSRT [16], Padchest [17], Montgomery [18], and Shenzhen [19]), with 245, 137, 566 and 138 examples respectively. JSRT cases are from patients diagnosed with lung nodules, while Padchest contains patients with a cardiomegaly diagnosis and features 20 examples where a pacemaker occludes the lung border. These two datasets contain heart and lung contour ground truth labels and are combined in a single dataset of 382 examples. Montgomery and Shenzhen contain lung contour ground truth labels only, and are combined into a second dataset of 704 cases where 394 examples are from patients with tuberculosis and 310 are from patients without. Each combined dataset is randomly split into 70% train, 15% validation and 15% test examples, each with a 1024px  $\times$  1024px resolution X-ray image and ground truth point coordinates for organ contours obtained from [5].

#### 3.2 Model Implementation and Training

We implement our model in PyTorch and use PyTorch-Geometric for the graph layer. All models were trained for 2500 epochs using a NVIDIA A100 GPU from Queen Mary’s Andrena HPC facility. For reliable performance estimates, all models and baselines were trained from scratch three times, the mean scores obtained for quantitative analysis and the median model used for qualitative analysis. Hyperparameters for all experiments were unchanged from [1]. To impose a unit Gaussian prior on the VAE bottleneck we train the network with an additional KL-divergence loss term with weight  $1e^{-5}$ , and use  $\beta = 2.5e^{-2}$  for the HCD weight. For joint models we pretrain the first UNet model separately using the recipe from [1] and freeze its weights when training the full model. This is done to reduce complexity in our training procedure.

#### 3.3 Comparison to Existing Methods and Ablation Study

We compare our approach to a variety of different dense- and point-based segmentation methods. First we validate our joint DD-DG learning approach by comparing to a DD-only segmentation network (UNet [2]) and DG-only segmentation networks (HybridGNet [6], HybridGNet+ISGC [1]).



**Fig. 4.** JSRT & Padchest: Qualitative Analysis. Note that our method does not suffer from the topological errors of dense-based methods but benefits from their segmentation accuracy. Specifically, improvements (white boxes) are most prevalent in areas of complexity such as where the heart and lungs intersect.

Next, we explore five alternative configurations of our joint architecture to demonstrate that our design choices are superior. These are: (1) UNet Joint: a network that uses our joint learning strategy but with two DD (UNet) networks, (2) Hourglass: joint learning but with no sharing between DD decoder and DG encoder, only the output of DD is passed to the input of DG, similar to the stacked hourglass network [21, 22], (3) Hourglass Concat: as above, but the output of DD is concatenated with the input and both are passed to DG, (4) Multi-task: a single dense encoder is shared between a dense and graph decoder, similar to [23], (5) No Joint: our network with no joint learning strategy.

To demonstrate the effectiveness of our HCD loss, we compare to our joint network trained with the contour term removed (MSE only). Our HCD loss is similar to differentiable polygon rasterization in BoundaryFormer [9], as they both use the distance field to represent points with respect to the true boundary. However, our method precomputes the distance field for each example and samples it during training, while BoundaryFormer approximates it on the fly. Hence we also compare to a single DG network (HybridGNet+IGSC) where each point output is rendered to a dense  $1028\text{px} \times 1028\text{px}$  segmentation map using rasterization and the full model is trained using a pixel-level loss.

Tables 1 and 2 demonstrate that our methodology outperforms all point- and dense-based segmentation baselines on both datasets. As seen in Fig. 4, the performance increase from networks that combine image features from dense and point trained networks (column 7,9) is superior to when image features from two dense trained networks are combined (column 5). Furthermore, concatenating features at each encoder-decoder level (Tables 1 and 2, row 11) instead of at the input-output level (row 5–6) shows improved performance. The addition of HCD supervision to a DG model (Tables 1 and 2, row 8) gives similar improvements in segmentation when compared to using a differentiable rasterization pipeline (row 10), yet is far more computationally efficient (Table 2, column 7).

**Table 1.** JSRT & Padchest Dataset: Quantitative Analysis

	Predict	Supervision	Lungs		Heart			
			DC↑	HD↓	JC↑	DC↑	HD ↓	JC↑
HybridGNet	point	point	0.9313	17.0445	0.8731	0.9065	15.3786	0.8319
HybridGNet+IGSC	point	point	0.9589	13.9955	0.9218	0.9295	13.2500	0.8702
UNet	dense	dense	0.9665	28.7316	0.9368	0.9358	29.6317	0.8811
UNet Joint	dense	dense	0.9681	26.3758	0.9395	0.9414	24.9409	0.8909
Hourglass	point	both	0.9669	13.4225	0.9374	0.9441	12.3434	0.8954
Hourglass Concat	point	both	0.9669	13.5275	0.9374	0.9438	12.1554	0.8948
Multi-task	point	both	0.9610	15.0490	0.9257	0.9284	13.1997	0.8679
No Joint	point	point	0.9655	13.2137	0.9341	0.9321	13.1826	0.8748
MSE Only	point	both	0.9686	<b>12.4058</b>	0.9402	0.9439	12.0872	0.8953
Rasterize	point	dense	0.9659	13.7267	0.9349	0.9344	12.9118	0.8785
Ours	point	both	<b>0.9698</b>	13.2087	<b>0.9423</b>	<b>0.9451</b>	<b>11.7721</b>	<b>0.8975</b>

**Table 2.** Montgomery & Shenzhen Dataset: Quantitative Analysis + Inference Time

	Predict	Supervision	DC↑	HD↓	JC↑	Inference (s)
HybridGNet	point	point	0.9459	12.0294	0.8989	0.0433
HybridGNet+IGSC	point	point	0.9677	9.7591	0.9380	0.0448
UNet	dense	dense	0.9716	16.7093	0.9453	0.0047
UNet Joint	dense	dense	0.9713	16.5447	0.9447	0.0103
Hourglass	point	both	0.9701	10.9284	0.9434	0.1213
Hourglass Concat	point	both	0.9712	10.8193	0.9448	0.1218
Multi-task	point	both	0.9697	10.8615	0.9417	0.0535
No Joint	point	point	0.9701	9.8246	0.9424	0.0510
MSE Only	point	both	0.9729	9.6527	0.9474	0.1224
Rasterize	point	dense	0.9718	<b>9.4485</b>	0.9453	0.2421
Ours	point	both	<b>0.9732</b>	10.2166	<b>0.9481</b>	0.1226

## 4 Conclusion

We proposed a novel segmentation architecture which leverage the benefits of both dense- and point- based algorithms to improve accuracy while reducing topological errors. Extensive experiments support our hypothesis that networks that utilise joint dense-point representations can encode more discriminative features which are both semantically rich and highly detailed. Limitations in segmentation methods using a point-wise distance were identified, and remedied with a new contour-aware loss function that offers an efficient alternative to differentiable rasterization methods. Our methodology can be applied to any graph segmentation network with a convolutional encoder that is optimised using a point-wise loss, and our experiments across four datasets demonstrate that our approach is generalizable to new data.

**Acknowledgements.** This research is part of AI-based Cardiac Image Computing (AICIC) funded by the faculty of Science and Engineering at Queen Mary University of London.

## References

1. Gaggion, N., Mansilla, L., Mosquera, C., Milone, D.H., Ferrante, E.: Improving anatomical plausibility in medical image segmentation via hybrid graph neural networks: applications to chest x-ray analysis. *IEEE Trans. Med. Imaging* (2022)
2. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015, Part III. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
3. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)
5. Gaggion, N., Vakalopoulou, M., Milone, D.H., Ferrante, E.: Multi-center anatomical segmentation with heterogeneous labels via landmark-based models. In: IEEE 20th International Symposium on Biomedical Imaging (2023)
6. Gaggion, N., Mansilla, L., Milone, D.H., Ferrante, E.: Hybrid graph convolutional neural networks for landmark-based anatomical segmentation. In: de Bruijne, M., et al. (eds.) MICCAI 2021. LNCS, vol. 12901, pp. 600–610. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-87193-2\\_57](https://doi.org/10.1007/978-3-030-87193-2_57)
7. Ling, H., Gao, J., Kar, A., Chen, W., Fidler, S.: Fast interactive object annotation with Curve-GCN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5257–5266 (2019)
8. Loper, M.M., Black, M.J.: OpenDR: an approximate differentiable renderer. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 154–169. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10584-0\\_11](https://doi.org/10.1007/978-3-319-10584-0_11)
9. Lazarow, J., Xu, W., Tu, Z.: Instance segmentation with mask-supervised polygonal boundary transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4382–4391 (2022)
10. Gur, S., Shaharabany, T., Wolf, L.: End to end trainable active contours via differentiable rendering. In: International Conference on Learning Representations (2020)
11. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
12. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: 2nd International Conference on Learning Representations, ICLR, Banff, Canada, 14–16 April 2014, Conference Track Proceedings (2014)
13. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
14. PyTorch Geometric: Cheb Conv Module. <https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html>. Accessed 7 Feb 2023

15. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 679–698 (1986)
16. Shiraishi, J.: Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *Am. J. Roentgenol.* **174**(1), 71–74 (2000)
17. Bustos, A., Pertusa, A., Salinas, J.M., de la Iglesia-Vayá, M.: PadChest: a large chest x-ray image dataset with multi-label annotated reports. *Med. Image Anal.* **66**, 101797 (2020)
18. Candemir, S., et al.: Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE Trans. Med. Imaging* **33**(2), 577–590 (2014)
19. Jaeger, S., et al.: Automatic tuberculosis screening using chest radiographs. *IEEE Trans. Med. Imaging* **33**(2), 233–245 (2014)
20. King, T., Butcher, S., Zalewski, L.: Apocrita - high performance computing cluster for Queen Mary University of London, Zenodo (2017). <https://doi.org/10.5281/zenodo.438045>
21. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46484-8\\_29](https://doi.org/10.1007/978-3-319-46484-8_29)
22. Xu, T., Takano, W.: Graph stacked hourglass networks for 3D human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16105–16114 (2021)
23. Li, W., Zhao, W., Zhong, H., He, C., Lin, D.: Joint semantic-geometric learning for polygonal building segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 3, pp. 1958–1965 (2021)