



From Astronomy to Histology: Adapting the FellWalker Algorithm to Deep Nuclear Instance Segmentation

Michael Yeung^{1,2} , Todd Watts³ , and Guang Yang²

¹ School of Clinical Medicine, University of Cambridge, Cambridge, UK

² National Heart and Lung Institute, Imperial College London, London, UK
michael.yeung21@imperial.ac.uk

³ Department of Medical Genetics, University of Cambridge, Cambridge, UK

Abstract. Accurate cell nuclei segmentation is necessary for subsequent histopathology image analysis, including tumour classification, grading and prognosis. Manually identifying cell nuclei is both difficult and time-consuming, with cell nuclei exhibiting dramatic differences in morphology and staining characteristics. Recently, significant advancements in automatic cell nuclei segmentation have been achieved using deep learning, with methods particularly successful in identifying cell nuclei from background tissue. However, delineating individual cell nuclei remains challenging, with often unclear boundaries between neighbouring nuclei. In this paper, we incorporate the FellWalker algorithm, originally developed for analysing molecular clouds, into a deep learning-based pipeline to perform instance cell nuclei segmentation. We evaluate our proposed method on the Lizard dataset, the largest publicly available nuclear segmentation dataset in digital pathology, and compare it against popular methods such as U-Net with Watershed and Mask R-CNN. Our proposed method consistently outperforms the other methods across dataset sizes, achieving an object Dice of 0.7876, F1 score of 0.8245 and Aggregated Jaccard Index of 0.6526. The flexible nature of our pipeline incorporating the FellWalker algorithm has the potential for broader application in biomedical image instance segmentation tasks.

Keywords: Digital pathology · Deep learning · Image processing · Instance segmentation

1 Introduction

Histopathology examination remains the gold standard for determining cancer diagnosis and prognosis. While the demand for biopsy sample analysis continues to increase, the number of pathologists remain in short supply [34]. Digital pathology is an important advancement that involves automating the process of analysing histopathology slides [18], with comparable performance to humans for detecting tumours [2, 25, 27]. To localise, quantify and characterise cells for diagnosis, accurate cell nuclei segmentation is required.

Classical approaches to automating cell nuclei segmentation include Otsu thresholding [32], clustering [20], graph-based [41], and active contour techniques [8]. However, the reliance of these methods on clear pixel intensity differences limits their use in identifying cell nuclei, which exhibit significant variation in morphology and staining on histology slides. Furthermore, these methods are unable to distinguish between overlapping nuclei, and often depend on first applying a distance transform to the segmentation mask [1], followed by processing using a watershed algorithm [33, 43, 45]. This task is known as instance segmentation, which is more challenging than the pixel-level classification seen in semantic segmentation, because an additional object detection stage is required.

Later approaches focused on applying traditional machine learning techniques [3], and in recent years, significant improvements have been achieved by adopting deep learning methods [16]. Deep neural networks learn complex representations combining shape, colour and texture information, and with the introduction of Fully Convolutional Networks (FCN), enabled Convolutional Neural Network (CNN) architectures to handle semantic image segmentation tasks [26]. Current state-of-the-art approaches for nuclear segmentation are largely based on the U-Net, a modified FCN architecture comprised of symmetrical encoder-decoder subnetworks with skip connections [35].

The simplest method to adapt the U-Net for instance segmentation involves directly learning ternary maps, training the model to predict separate labels for the nucleus, background and boundary pixels [7, 23]. More complicated segmentation pipelines such as Cellpose [39] and Stardist [36], use a U-Net variant to indirectly predict auxiliary representations of segmentation masks, in these cases gradient vector fields and star-convex polygons respectively.

Mask R-CNN [14] is another CNN-based innovation originally developed for natural image segmentation, but has also been applied to nuclear segmentation by frameworks such as nucleAIzer [17]. Here, Mask R-CNN uses a regional proposal network to first identify possible cell nuclei regions, which are then processed as a binary segmentation task to identify individual nuclei. Newer methods have combined Mask-RCNN with the U-Net, either in series [19], as an ensemble [42], or through merging the two methods by integrating the regional proposal network into the U-Net architecture [44].

Recent approaches have also focused on integrating deep learning with traditional methods, leveraging the representational power of CNNs with the simplicity of traditional methods to reduce inference times. Separate U-Nets have been used to output the segmentation mask and distance transform map, which were then processed using a watershed algorithm [29]. This has also been accomplished using a single modified U-Net which outputs both the segmentation mask and distance transform map, which is not only more parameter efficient, but may also offer additional performance benefits through regularisation [28].

Despite significant performance gains observed when combining deep learning with watershed, the segmentation accuracy is limited by the reliance of the watershed algorithm on clear pixel intensity differences at nuclei borders. The FellWalker algorithm is related to watershed methods, but its use has been

largely restricted to analysing molecular clouds in astronomy [4]. Coincidentally, radio observations of molecular clouds resemble distance transform maps in appearance, and applying the FellWalker algorithm on this auxiliary representation of the segmentation mask avoids relying on the original image to identify boundaries.

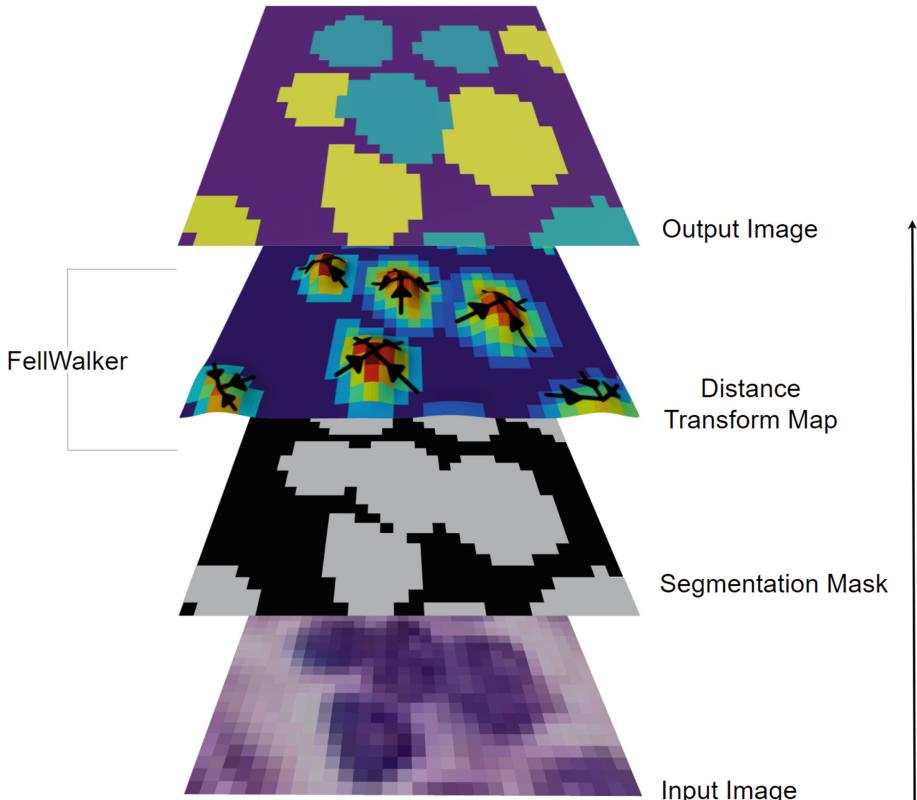


Fig. 1. The FellWalker algorithm applied to nuclear instance segmentation. Each pixel identified as a cell nuclei in the segmentation mask is used as a starting point for a walk on the distance transform map, which follows an uphill path (arrows) by moving to the neighbourhood pixel with the highest intensity as the next step in the walk. Pixels are clustered together if they converge onto the same peak, producing the output image.

The main contributions of this work are summarised as follows:

1. We adapt the FellWalker algorithm from analysing molecular clouds in astronomy, to instance segmentation of cell nuclei in histology.
2. We evaluate our method on the Lizard dataset against the U-Net with Watershed and Mask R-CNN methods.

3. We demonstrate further performance gains by evaluating several other deep learning architectures used to generate the segmentation mask and distance transform map.

2 Materials and Methods

2.1 Dataset Description and Evaluation Metrics

For our experiments, we use the Lizard dataset [12], which combines data from 6 separate data sources and is the largest instance segmentation and classification dataset in digital pathology. This dataset consists of whole slide images at 20x objective magnification of colonic tissue stained with Haematoxylin and Eosin. In total, 495,179 nuclei with manually refined labels are provided, classified as either: epithelial cell, lymphocyte, plasma cell, neutrophil, eosinophil or connective tissue.

For evaluation, we use three metrics from the MICCAI 2015 Gland Segmentation in Colon Histology Images Challenge Contest (GlaS) and MICCAI 2018 Multi-Organ Nucleus Segmentation Challenge (MoNuSeg): the object-level Dice index (Dice_{obj}) for segmentation accuracy, the F1 score metric (F1) for detection accuracy, and the average Aggregated Jaccard Index (AJI) which incorporates both segmentation and detection accuracy [23, 24, 37, 38].

The Dice index is a common overlap metric for evaluating pixel-level segmentation accuracy [11], defined as:

$$\text{Dice} = \frac{2|X \cap Y|}{|X| + |Y|}, \quad (1)$$

where X and Y represent the set of pixels labelled by the prediction mask and ground truth respectively. To account for the segmentation quality of individual objects, the Dice_{obj} is computed as:

$$\text{Dice}_{\text{obj}} = \frac{1}{2} \left[\sum_{i=1}^{n_Y} w_i \text{Dice}(Y_i, X_*(Y_i)) + \sum_{j=1}^{n_X} \tilde{w}_j \text{Dice}(Y_*(X_j), X_j) \right], \quad (2)$$

$$w_i = |Y_i| / \sum_{p=1}^{n_Y} |Y_p|, \quad \tilde{w}_j = |X_j| / \sum_{q=1}^{n_X} |X_q|, \quad (3)$$

where n_X and n_Y refer to the number of predicted and ground truth objects respectively, $X_*(Y_i)$ and $Y_*(X_j)$ refer to the ground truth object with maximum overlap with predicted object i, and predicted object with maximum overlap with ground truth object j, respectively, and w_i and \tilde{w}_j are weights used to more heavily penalise the prediction of larger ground truth objects.

The F1 score measures the detection accuracy, with a predicted object that has at least 50% overlap with a ground truth object assigned as a true positive (TP), and a false positive (FP) otherwise. The number of false negative (FN)

predictions is given by subtracting the number of true positives from number of ground truth objects. The F1 score is defined as:

$$F1 = \frac{2TP}{2TP + FP + FN}. \quad (4)$$

Finally, the AJI encompasses both segmentation and detection accuracy, and involves matching every ground truth object with a detected object that has maximal overlap. The AJI is computed as the ratio of the sum of the cardinals of intersection and union of these matched objects, with the remaining detected objects added to the denominator.

The value of all three metrics range from 0 to 1, with a higher value indicating better performance.

2.2 FellWalker Algorithm

The FellWalker algorithm was originally developed for clustering radio observations of molecular clouds [4]. The algorithm belongs to a larger class of watershed algorithms, which consider grayscale images as topographic maps, where height is represented by pixel intensity, to separate the image into regions based on local minima [22]. However, the FellWalker algorithm differs from other watershed algorithms by instead using gradient information and local maxima to segment image regions. The key method is summarised in Algorithm 1.

Following the fell-walking metaphor, which refers to the pastime of hill walking, the algorithm involves iteratively taking uphill walks from various pixel starting locations, and following the direction of steepest ascent by querying immediate neighbouring pixels. Therefore, low-intensity pixels follow paths towards a peak, which represents the highest pixel intensity within its broader neighbouring region. Any pixels encountered on a walk towards the same peak are assigned to the same image region (Fig. 1).

2.3 Deep Learning-Based Nuclear Segmentation

The overview of the pipeline is shown in Fig. 2.

The image is first processed by a deep neural network based on the U-Net architecture [35], but with two separate decoder pathways which are trained to produce either the segmentation mask or the distance transform map. The distance transform map is computed using the Euclidean distance and normalised in the range $[0, 1]$. Both decoder pathways end with a final sigmoid activation function.

Prior to applying the FellWalker algorithm, the segmentation mask is used to mask out the background pixels in the distance transform map.

2.4 Implementation Details

Our experiments were carried out on Google Colab using Tesla P100 GPUs. We used image patches of the Lizard dataset provided by the CoNIC challenge

Algorithm 1: FellWalker algorithm

```

Input : 2D array of data  $D$ , mask  $M$  and threshold  $T$ 
Output: 2D array of assigned clusters  $C$ 

1 Initialise output array  $C \leftarrow$  array of zeros with same shape as  $D$ ;
2 Initialise path list  $P \leftarrow []$ ;
3 Initialise cluster value  $N \leftarrow 1$ ;
4 for pixel in  $D$  do
5   if pixel in  $M$  and pixel value >  $T$  and pixel not in P then
6      $path = []$ ;
7     while True do
8       append pixel to  $path$ ;
9       if pixel value higher than immediate neighbourhood pixel values then
10         if pixel value higher than broader neighbourhood pixel values then
11           append  $path$  to  $P$ ;
12           break;
13         end
14       end
15     else
16       if highest immediate neighbourhood pixel value in P then
17         append  $path$  to same list as immediate neighbourhood pixel
18         in  $P$ ;
19         break;
20       end
21     else
22       |  $pixel \leftarrow$  highest immediate neighbourhood pixel value;
23     end
24   end
25 end
26 end
27 for path in  $P$  do
28   for pixel in path do
29     |  $pixel$  in  $C \leftarrow N$ 
30   end
31    $N \leftarrow N + 1$ ;
32 end

```

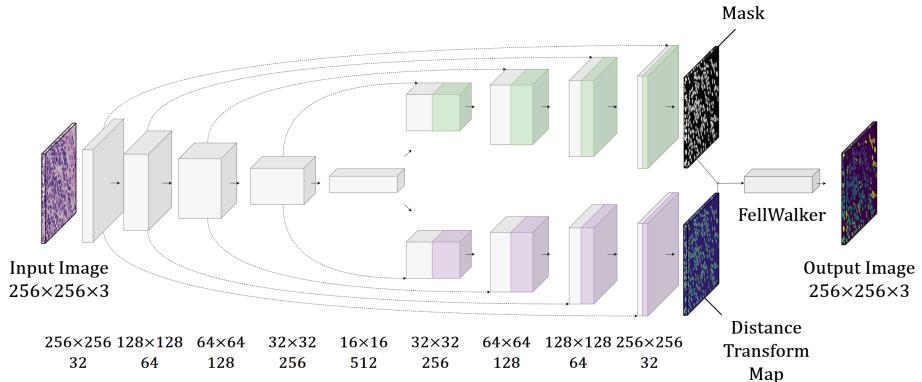


Fig. 2. Overview of our proposed method. The modified U-Net processes an input image through two decoder pathways in parallel to learn a segmentation mask and distance transform map. The segmentation mask is used to mask background pixels in the distance transform map prior to input into the FellWalker algorithm to produce the final instance segmentation output image.

2022, which consists of 4981 non-overlapping image patches of size 256×256 [13]. We randomly partitioned the dataset into 80% development set and 20% test set, and further partitioned the development set into 80% training and 20% validation set. We trained models on different subsets of the development set used as shown in Table 1.

Table 1. Number of image patches used for training, validation and testing. The same test patches are used for each dataset proportion.

% Data	#Train	#Validation	#Test
1	30	8	969
10	309	78	969
25	774	194	969
50	1549	387	969
100	3098	774	969

Image patches were normalised to $[0, 1]$, and we applied the following on-the-fly data augmentation using the Albumentations library [6], with $p = 0.2$: crop, flip (horizontal and vertical), rotation ($90^\circ, 180^\circ, 270^\circ$), CLAHE, gamma, brightness, contrast and blur.

We used OpenCV's implementation of the Meyer's flooding algorithm (referred to as Watershed) [30], and CUPID's implementation of the FellWalker algorithm [5]. For fair comparison, we applied these algorithms on the same binary segmentation masks and distance transform maps produced by the deep neural networks.

To evaluate the performance of the FellWalker algorithm using different deep neural networks, we used a range of modified state-of-the-art neural networks with ResNet-50 pretrained encoders: U-Net [35], LinkNet [9], DeepLabv3+ [10] and U-Net++ [47]. All neural networks were implemented in PyTorch, except for the Mask R-CNN which was implemented in Keras with Tensorflow backend.

We initialised the models using He initialisation [15], and trained each model with Batch normalisation, using the Adam optimiser with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [21]. We used a batch size of 8 and initial learning rate of 0.001. For convergence criteria, we reduced the learning rate by 0.1 if the validation loss did not improve after 5 epochs, and terminated training if the validation loss did not improve after 15 epochs.

To train the neural networks, we used the Dice loss ($\mathcal{L}_{\text{Dice}}$) and binary cross entropy loss (\mathcal{L}_{BCE}) for binary segmentation [40, 46], and the mean squared error (\mathcal{L}_{MSE}) for the distance transform map regression:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{BCE}} + \lambda \mathcal{L}_{\text{MSE}}, \quad (5)$$

where

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{1}{n} \sum_{i=1}^n \text{Dice}(X_i, Y_i), \quad (6)$$

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n} \sum_{i=1}^n (Y_i \log(X_i) + (1 - Y_i) \log(1 - X_i)), \quad (7)$$

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (Y_i - X_i)^2. \quad (8)$$

Here, λ control the relative contribution of the losses and is set as 10 to maintain a significant contribution from the regression loss.

To test for statistical significance, we used the Wilcoxon rank sum test. A statistically significant difference was defined as $p < 0.05$.

3 Results

In this section, we first describe the results comparing three different algorithms across a range of dataset sizes, followed by a comparison of five state-of-the-art deep neural networks and the effect on the FellWalker algorithm performance.

3.1 Algorithm Performances Across Different Dataset Sizes

The results for the performance of the Watershed, Mask R-CNN and FellWalker algorithms on various dataset sizes of the Lizard dataset are shown in Table 2, and visualised in Fig. 3. Across all dataset sizes, the best performance was consistently observed with the FellWalker algorithm, achieving a Dice_{obj} of 0.7876 ± 0.0041 , F1 score of 0.8245 ± 0.0053 and AJI of 0.6526 ± 0.0052 using 100% of the data. Moreover, the FellWalker algorithm trained on 10%

of the data outperformed the Watershed and Mask R-CNN trained on 100% of the data. The Watershed was the lowest performing algorithm across the dataset sizes. All algorithms demonstrated better performance with larger training data, but only limited performance gains were observed when using over 10% of the data. For qualitative comparison, example segmentations are shown in Fig. 4.

Table 2. Performance comparisons using the Watershed, Mask R-CNN and FellWalker algorithms on different proportions of the Lizard dataset. The highest values for each dataset size is indicated in bold. The 95% confidence intervals are displayed in brackets. *Statistically significant difference ($p < 0.05$).

%Data	Algorithm	Dice _{obj} (\uparrow)	F1 (\uparrow)	AJI (\uparrow)
1	Watershed	0.3716 (0.0091)	0.3523 (0.0137)	0.2506 (0.0088)
	Mask R-CNN	0.5170 (0.0081)	0.5912* (0.0108)	0.3487 (0.0078)
	FellWalker	0.5332* (0.0074)	0.5674 (0.0120)	0.3786* (0.0069)
10	Watershed	0.4957 (0.0056)	0.4598 (0.0089)	0.4138 (0.0042)
	Mask R-CNN	0.6299 (0.0057)	0.7140 (0.0079)	0.4681 (0.0060)
	FellWalker	0.7252* (0.0044)	0.7516* (0.0066)	0.5708* (0.0055)
25	Watershed	0.5243 (0.0054)	0.5035 (0.0086)	0.4359 (0.0040)
	Mask R-CNN	0.6650 (0.0054)	0.7404 (0.0072)	0.5034 (0.0059)
	FellWalker	0.7541* (0.0041)	0.7868* (0.0058)	0.6077* (0.0052)
50	Watershed	0.5417 (0.0052)	0.5309 (0.0082)	0.4509 (0.0040)
	Mask R-CNN	0.6978 (0.0047)	0.7713 (0.0060)	0.5275 (0.0059)
	FellWalker	0.7777* (0.0041)	0.8126* (0.0055)	0.6389* (0.0051)
100	Watershed	0.5498 (0.0051)	0.5432 (0.0081)	0.4578 (0.0038)
	Mask R-CNN	0.7219 (0.0045)	0.7896 (0.0061)	0.5630 (0.0056)
	FellWalker	0.7876* (0.0041)	0.8245* (0.0053)	0.6526* (0.0052)

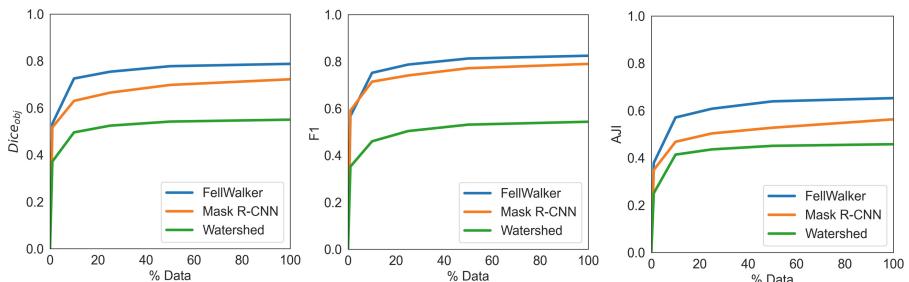


Fig. 3. Graphical representation of the performances of the Watershed, Mask R-CNN and FellWalker algorithms trained on different proportions of the Lizard dataset.

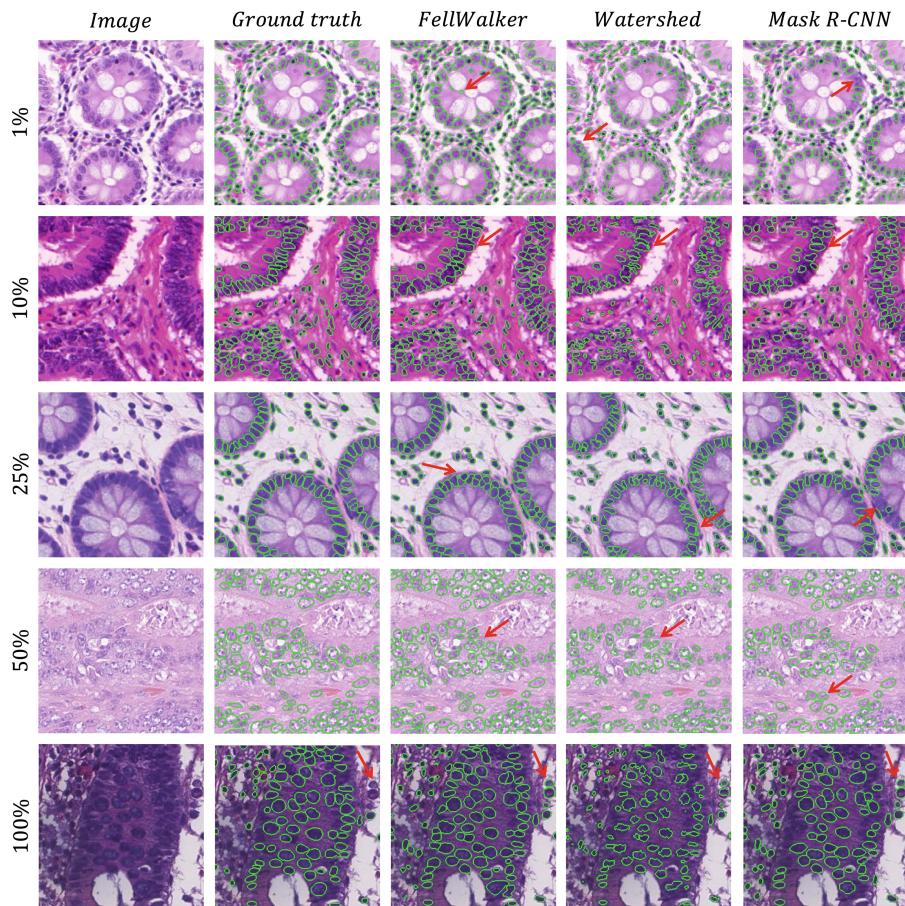


Fig. 4. Example segmentations for each algorithm on each dataset proportion. The image and ground truth are provided for reference. The red arrows highlight example areas where segmentation quality differs. (Color figure online)

Across dataset sizes, the highest quality segmentations were generally produced by the FellWalker algorithm, although a higher number of false positive predictions are visible at smaller dataset sizes when compared to Watershed algorithm and Mask R-CNN. In contrast, the Mask R-CNN appears to exhibit a lower true positive rate and tendency to undersegment the nuclei. The Watershed algorithm appears to consistently undersegment the cell nuclei irrespective of dataset size, producing irregular borders.

The inference times for the three algorithms are shown in Table 3. The Watershed algorithm performed the fastest inference, followed by the FellWalker algorithm and Mask R-CNN.

Table 3. Inference time for the Watershed, Mask R-CNN and FellWalker algorithms. The highest value is indicated in bold. The 95% confidence intervals are displayed in brackets. *Statistically significant difference ($p < 0.05$).

Algorithm	Inference time (s) (\downarrow)
Watershed	0.013* (0.006)
Mask R-CNN	0.450 (0.012)
FellWalker	0.190 (0.011)

3.2 FellWalker Performance with Different Segmentation Models

The results of applying the FellWalker algorithm to outputs from various state-of-the-art deep neural networks on 1% of the Lizard dataset are shown in Table 4, and visualised in Fig. 5. Generally, larger networks performed better, with the U-Net++ achieving the highest $Dice_{obj}$ of 0.6553 ± 0.0053 , F1 score of 0.7080 ± 0.0085 and AJI of 0.4932 ± 0.0059 .

Table 4. Performance comparisons of different deep neural networks with the FellWalker algorithm on 1% of the Lizard dataset. The highest value is indicated in bold. The 95% confidence intervals are displayed in brackets. *Statistically significant difference ($p < 0.05$).

Model	Backbone	#Params (10^6)	$Dice_{obj}$ (\uparrow)	F1 (\uparrow)	AJI (\uparrow)
U-Net [35]	-	10.8	0.5332 (0.0074)	0.5674 (0.0120)	0.3786 (0.0069)
DeepLabv3+ [10]	ResNet-50	26.8	0.5710 (0.0057)	0.6349 (0.0092)	0.4205 (0.0054)
LinkNet [9]	ResNet-50	38.8	0.6318 (0.0055)	0.6751 (0.0085)	0.4718 (0.0058)
U-Net [35]	ResNet-50	64.3	0.6458 (0.0061)	0.6723 (0.0092)	0.4886 (0.0064)
U-Net++ [47]	ResNet-50	99.1	0.6553* (0.0053)	0.7080* (0.0085)	0.4932* (0.0059)

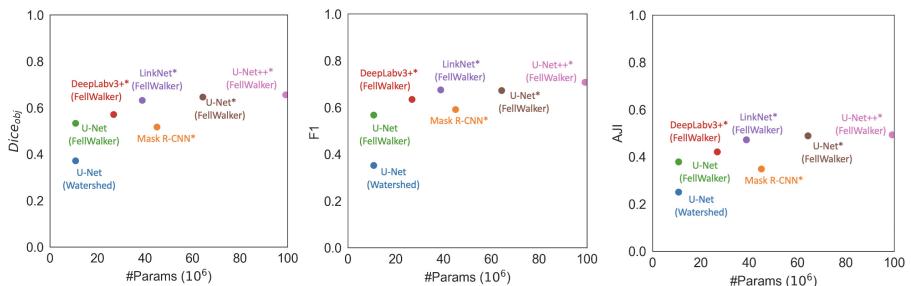


Fig. 5. Graphical representation of the performances of different neural networks with the FellWalker on 1% of the Lizard dataset. The performances of the Watershed and Mask R-CNN algorithms are shown for reference. *models with ResNet-50 pretrained encoders.

4 Discussion and Conclusions

In this work, we incorporated the FellWalker algorithm as part of a deep learning-based pipeline to perform automatic instance segmentation of cell nuclei on histology slides. We observed significantly improved performance using the FellWalker algorithm compared to the Watershed and Mask R-CNN algorithms, consistent across training data subsets of different sizes, from 1% to 100% of the Lizard dataset. Moreover, we compared the performance of the FellWalker algorithm using different deep neural network outputs, and demonstrated significant performance improvements with pretrained encoders and more complex networks.

Two reasons explain the performance improvements observed when using the FellWalker algorithm over the Watershed algorithm. Firstly, the FellWalker algorithm relies heavily on the segmentation mask to select the pixels used for clustering. Identifying cell nuclei among background tissue is a relatively easy task for deep neural networks to handle, and therefore serves as a useful prior for the FellWalker algorithm to exploit. In contrast, the Watershed algorithm uses the segmentation mask to establish the background tissue, but does not guarantee that identified cell nuclei regions will be retained, causing it to under-segment across dataset sizes (Fig. 4). Secondly, the FellWalker algorithm is a gradient-tracing method and is therefore data-efficient with respect to the distance transform map. This differs from the Watershed algorithm, which thresholds the distance transform map to establish markers, resulting in the loss of gradient information. This gradient information is particularly useful where pixel intensities from the image are less informative, which is often the case with histology slides where heterogeneous staining is a significant issue.

The watershed algorithm offers two advantages over the FellWalker algorithm, despite its lower segmentation accuracy. Firstly, the reliance of the FellWalker algorithm on the segmentation mask reduces robustness to error from the segmentation network. This is apparent with more false positive predictions visible compared to the Watershed algorithm when trained on 1% of the data (Fig. 4). However, the robustness of the FellWalker algorithm to segmentation noise can be tuned with the threshold hyperparameter. Secondly, the Watershed algorithm is significantly faster than the FellWalker algorithm (Table 3). Although this is less relevant in the context of analysing histopathology slides where real-time inference is not required and performance is more important.

In future work, we will further investigate different deep neural network architectures as input to the FellWalker algorithm. We used entirely separate decoder pathways for the segmentation mask and distance transform map, but other variants such as multi-head and reconstruction branches are possible [28]. Furthermore, it would be useful to incorporate other image features such as edge information to improve the performance of the FellWalker algorithm [31]. Finally, implementation of the FellWalker algorithm for 3D data is available [5], and therefore suggests the potential for wider application in biomedical image instance segmentation tasks.

Acknowledgements. This work was funded in part by the UK Research and Innovation Future Leaders Fellowship [MR/V023799/1], in part by the Medical Research Council [MC/PC/21013], in part by the European Research Council Innovative Medicines Initiative [DRAGON, H2020-JTI-IMI2 101005122], and in part by the AI for Health Imaging Award [CHAIMELEON, H2020-SC1-FA-DTS2019-1 952172]. The views expressed are those of the authors and not necessarily those of the NHS, the NIHR, or the Department of Health and Social Care.

References

1. Abdolhoseini, M., Kluge, M.G., Walker, F.R., Johnson, S.J.: Segmentation of heavily clustered nuclei from histopathological images. *Sci. Rep.* **9**(1), 1–13 (2019)
2. Alsubaie, N., Sirinukunwattana, K., Raza, S.E.A., Snead, D., Rajpoot, N.: A bottom-up approach for tumour differentiation in whole slide images of lung adenocarcinoma. In: *Medical Imaging 2018: Digital Pathology*, vol. 10581, p. 105810E. International Society for Optics and Photonics (2018)
3. Berg, S., et al.: Ilastik: interactive machine learning for (bio) image analysis. *Nat. Methods* **16**(12), 1226–1232 (2019)
4. Berry, D.S.: Fellwalker - a clump identification algorithm. *Astron. Comput.* **10**, 22–31 (2015)
5. Berry, D., Reinholt, K., Jenness, T., Economou, F.: Cupid: a clump identification and analysis package. In: *Astronomical Data Analysis Software and Systems XVI*, vol. 376, p. 425 (2007)
6. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: fast and flexible image augmentations. *Information* **11**(2), 125 (2020)
7. Caicedo, J.C., et al.: Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry A* **95**(9), 952–965 (2019)
8. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
9. Chaurasia, A., Culurciello, E.: Linknet: Exploiting encoder representations for efficient semantic segmentation. In: *2017 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4. IEEE (2017)
10. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818 (2018)
11. Dice, L.R.: Measures of the amount of ecologic association between species. *Ecology* **26**(3), 297–302 (1945)
12. Graham, S., et al.: Lizard: a large-scale dataset for colonic nuclear instance segmentation and classification. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 684–693 (2021)
13. Graham, S., et al.: Conic: Colon nuclei identification and counting challenge 2022. arXiv preprint [arXiv:2111.14485](https://arxiv.org/abs/2111.14485) (2021)
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034 (2015)
16. Hollandi, R., Moshkov, N., Paavolainen, L., Tasnadi, E., Piccinini, F., Horvath, P.: Nucleus segmentation: towards automated solutions. *Trends Cell Biol.* (2022)

17. Hollandi, R., et al.: Nucleaizer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Syst.* **10**(5), 453–458 (2020)
18. Irshad, H., Veillard, A., Roux, L., Racoceanu, D.: Methods for nuclei detection, segmentation, and classification in digital histopathology: a review-current status and future potential. *IEEE Rev. Biomed. Eng.* **7**, 97–114 (2013)
19. Jung, H., Lodhi, B., Kang, J.: An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images. *BMC Biomed. Eng.* **1**(1), 1–12 (2019)
20. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 881–892 (2002)
21. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
22. Kornilov, A.S., Safonov, I.V.: An overview of watershed algorithm implementations in open source libraries. *J. Imaging* **4**(10), 123 (2018)
23. Kumar, N., et al.: A multi-organ nucleus segmentation challenge. *IEEE Trans. Med. Imaging* **39**(5), 1380–1391 (2019)
24. Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., Sethi, A.: A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Trans. Med. Imaging* **36**(7), 1550–1560 (2017)
25. Liu, Y., et al.: Detecting cancer metastases on gigapixel pathology images. arXiv preprint [arXiv:1703.02442](https://arxiv.org/abs/1703.02442) (2017)
26. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)
27. Lu, C., et al.: A prognostic model for overall survival of patients with early-stage non-small cell lung cancer: a multicentre, retrospective study. *Lancet Digit. Health* **2**(11), e594–e606 (2020)
28. Ma, J., et al.: How distance transform maps boost segmentation CNNs: an empirical study. In: Medical Imaging with Deep Learning, pp. 479–492. PMLR (2020)
29. Mahbod, A., Schaefer, G., Ellinger, I., Ecker, R., Smedby, Ö., Wang, C.: A two-stage U-net algorithm for segmentation of nuclei in H&E-stained tissues. In: Reyes-Aldasoro, C.C., Janowczyk, A., Veta, M., Bankhead, P., Sirinukunwattana, K. (eds.) ECDP 2019. LNCS, vol. 11435, pp. 75–82. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23937-4_9
30. Meyer, F., Maragos, P.: Multiscale morphological segmentations based on watershed, flooding, and Eikonal PDE. In: Nielsen, M., Johansen, P., Olsen, O.F., Weickert, J. (eds.) Scale-Space 1999. LNCS, vol. 1682, pp. 351–362. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48236-9_31
31. Murugesan, B., Sarveswaran, K., Shankaranarayana, S.M., Ram, K., Joseph, J., Sivaprasam, M.: PSI-net: shape and boundary aware joint multi-task deep network for medical image segmentation. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 7223–7226. IEEE (2019)
32. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
33. Plissiti, M.E., Nikou, C., Charchanti, A.: Watershed-based segmentation of cell nuclei boundaries in pap smear images. In: Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine, pp. 1–4. IEEE (2010)

34. Robboy, S.J., et al.: Pathologist workforce in the united states: I. Development of a predictive model to examine factors influencing supply. *Arch. Pathol. Lab. Med.* **137**(12), 1723–1732 (2013)
35. 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. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
36. Schmidt, U., Weigert, M., Broaddus, C., Myers, G.: Cell detection with star-convex polygons. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11071, pp. 265–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00934-2_30
37. Sirinukunwattana, K., et al.: Gland segmentation in colon histology images: the GLAS challenge contest. *Med. Image Anal.* **35**, 489–502 (2017)
38. Sirinukunwattana, K., Snead, D.R., Rajpoot, N.M.: A stochastic polygons model for glandular structures in colon histology images. *IEEE Trans. Med. Imaging* **34**(11), 2366–2378 (2015)
39. Stringer, C., Wang, T., Michaelos, M., Pachitariu, M.: Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**(1), 100–106 (2021)
40. Taghanaki, S.A., et al.: Combo loss: handling input and output imbalance in multi-organ segmentation. *Comput. Med. Imaging Graph.* **75**, 24–33 (2019). <https://doi.org/10.1016/j.compmedimag.2019.04.005>
41. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
42. Vuola, A.O., Akram, S.U., Kannala, J.: Mask-RCNN and U-net ensembled for nuclei segmentation. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 208–212. IEEE (2019)
43. Wählby, C., Sintorn, I.M., Erlandsson, F., Borgefors, G., Bengtsson, E.: Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *J. Microsc.* **215**(1), 67–76 (2004)
44. Yang, L., et al.: NuSet: a deep learning tool for reliably separating and analyzing crowded cells. *PLoS Comput. Biol.* **16**(9), e1008193 (2020)
45. Yang, X., Li, H., Zhou, X.: Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and Kalman filter in time-lapse microscopy. *IEEE Trans. Circuits Syst. I Regul. Pap.* **53**(11), 2405–2414 (2006)
46. Yeung, M., Sala, E., Schönlieb, C.B., Rundo, L.: Unified Focal loss: generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Comput. Med. Imaging Graph.* **95**, 102026 (2022)
47. Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J.: UNet++: a nested U-net architecture for medical image segmentation. In: Stoyanov, D., et al. (eds.) DLMIA/ML-CDS -2018. LNCS, vol. 11045, pp. 3–11. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00889-5_1