Unsupervised Pre-training for Fully Convolutional Neural Networks

Stiaan Wiehman
CSIR-SU Centre for AI Research
Computer Science Division
Stellenbosch University
Stellenbosch, South Africa
Email: stiaan@aims.ac.za

Steve Kroon
CSIR-SU Centre for AI Research
Computer Science Division
Stellenbosch University
Stellenbosch, South Africa
Email: kroon@sun.ac.za

Hendrik de Villiers
Food and Biobased Research
Wageningen UR
Wageningen, The Netherlands
Email: hendrik.devilliers@wur.nl

Abstract—**Unsupervised pre-training of neural networks has been shown to act as a regularization technique, improving performance and reducing model variance. Recently, fully convolutional networks (FCNs) have shown state-of-the-art results on various semantic segmentation tasks. Unfortunately, there is no efficient approach available for FCNs to benefit from unsupervised pre-training. Given the unique property of FCNs to output segmentation maps, we explore a novel variation of unsupervised pre-training specifically designed for FCNs. We extend an existing FCN, called U-net, to facilitate end-to-end unsupervised pre-training and apply it on the ISBI 2012 EM segmentation challenge data set. We performed a battery of significance tests for both equality of means and equality of variance, and show that our results are consistent with previous work on unsupervised pre-training obtained from much smaller networks. We conclude that end-to-end unsupervised pre-training for FCNs adds robustness to random initialization, thus reducing model variance.**

I. Introduction

Unsupervised pre-training has been shown to have a regularization effect on multiple machine learning approaches [1]. Typical neural network approaches that employ unsupervised pre-training often involve stacks of autoencoders (and more recently, convolutional autoencoders [2]), which employ a certain form of unsupervised learning known as input reconstruction. The drawback in using autoencoders lies in the pre-training procedure. Each layer in a stacked (convolutional) autoencoder needs to be trained consecutively, which can become time-consuming, especially as the architecture becomes larger.

Unsupervised pre-training has not yet been applied to fully convolutional networks (FCNs), a recently developed class of neural networks which are mainly composed of convolutional layers, contain no fully connected layers and produce segmentation maps instead of single labels [3].

While FCNs could in principle employ traditional unsupervised pre-training approaches, building such a model using conventional convolutional autoencoders is not generally feasible, given the sheer depth of already established FCN model architectures. This paper aims to explore an alternative route of enabling unsupervised pre-training in FCNs, by expanding on an already established FCN known as U-net [4], which employs 23 convolutional layers.

Our approach rests on the ability of FCNs to output segmentation maps corresponding to (portions of) the original input: we propose a novel extension of FCNs achieving end-to-end autoencoding by having the *full model* reproduce the network input. The similarities of this approach to traditional unsupervised pre-training for neural networks motivates our use of the term in this paper. However, it is important to note that our approach actually combines the unsupervised and supervised aspects in a single training procedure with the focus shifting from the former to the latter during training. Furthermore, our current approach only employs the labeled data provided for regular training, unlike other unsupervised pre-training approaches.

We further explore the effect of our unsupervised pre-training approach on model performance, providing statistical evidence that the addition of unsupervised pre-training adds robustness to random initialization of the model weights.

II. Related Work

FCNs are an elegant neural network approach for performing semantic segmentation. The property that distinguishes them from conventional neural networks is that they are capable of producing segmentation maps as output, which makes them ideal for this study.

There have been a number of different applications of FCNs since their advent in Long et al. [3], including various tasks from bioimage domains. Ronneberger et al. [4] presented an FCN architecture called U-net, which consisted of a down-sampling pathway followed by an upsampling pathway. This model achieved state-of-the-art performance on the ISBI 2012 EM segmentation challenge. U-net was later outperformed by another FCN described by Chen et al. [5], which used multiple segmentation output layers at various points in the network. The aforementioned U-net architecture also won the ISBI 2015 cell tracking challenge, and forms the basis of this study. In our previous work [6], a similar architecture to U-net was shown to achieve state-of-the-art performance on the BBBC *C. elegans* live/dead assay data set. In Poudel et al. [7], a recurrent FCN was proposed that achieved state-of-the-art performance on two segmentation data sets, the MICCAI 2009 LV segmentation challenge and the PRETERM data set. Lastly, Chen et al. [8] utilized a 3D FCN to perform volumetric segmentation on three-dimensional magnetic resonance imaging data from the MICCAI 2015 Challenge on Automatic Intervertebral Disc Localization and Segmentation [9].

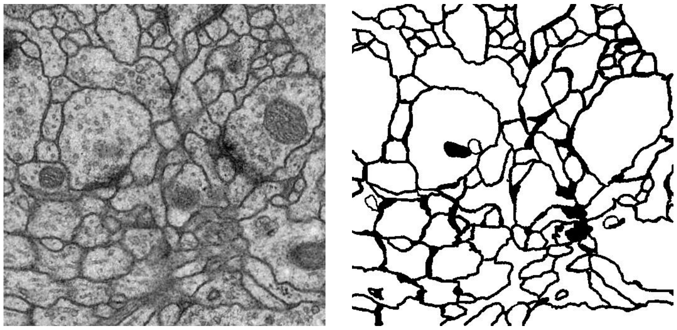
As mentioned earlier, FCNs present a unique opportunity to apply unsupervised pre-training due to their ability to produce output maps rather than single labels. The closest that unsupervised pre-training has come to FCN architectures, to the best of our knowledge, is stacked convolutional autoencoders, as defined by Mesci et al. [2]. A convolutional autoencoder is a convolutional layer that is required to reconstruct its input after applying a pooling operation over its feature maps (to discourage the trivial solution), and are typically trained using the standard greedy layer-wise approach.

Employing unsupervised pre-training followed by supervised fine-tuning can also be considered as semi-supervised learning. Conventional semi-supervised learning as it is used in neural networks often involve a data set of which only a small portion is labeled [1], [10], [11], [12]. There are a number of approaches that make use of semi-supervised learning, all of which have shown an improved performance over their purely supervised counterparts. In Hong et al. [10], a decoupled neural network is proposed that consists of two separate networks, one for classification and one for segmentation, connected by bridging layers. This approach represents a special instance of semi-supervised learning: one where the data set consists of weakly labeled data with a small portion of strongly labeled data.[1] In Kingma et al. [11], semi-supervised learning was used with deep generative models, showing state-of-the-art performance on the MNIST data set. In Rasmus et al. [12], unsupervised Ladder networks [13] were extended by adding a supervised learning component. Their resulting model reached state-of-the-art performance on both MNIST and CIFAR-10.

III. Data set

The data used in this study were obtained from the training data used in the ISBI 2012 EM segmentation challenge [14], [15]. The training data comprises thirty serial section transmission electron microscopy 512×512-pixel images showing the ventral nerve cord from a *Drosophila* larva. Each image depicts a number of cells separated by membranes; the task is segmenting the image (i.e. labelling each pixel as depicting either part of a cell or part of a membrane). For each image, a corresponding fully annotated ground truth segmentation map is provided, as illustrated in Figure 1. For the purpose of this work, the challenge training set was randomly divided into two sets of fifteen images each, with one used for training and the other for testing.

The images from the training set were sampled at random, followed by a random combination of transformations before they were used for training. These transformations were possible horizontal mirroring, rotations by multiples of 10°, and elastic deformations using parameters sampled from a continuous distribution[2]. This sampling ensures that any specific transformed image is extremely unlikely to reoccur during training, thus significantly reducing the risk of overfitting. It should also be noted that the chosen transformations were also applied to the ground truth mask, in order to obtain the correct pixel classifications corresponding to the transformed image. The original ground truth contained binary masks with pixels of value 0 and 255. Due to the interpolation performed during the transformations, this is no longer the case for the transformed ground truth. The label vector for each individual pixel is two-dimensional, the first representing membranes and the second, cells. To generate these label vectors, the transformed mask is divided into three regions and for each pixel value p at (i, j), the corresponding label vector $v(i, j)$ is then defined as


(a) Training Image

(b) Ground Truth

Fig. 1. An example image-label pair from the ISBI 2012 EM segmentation challenge. On the right, black pixels correspond to membranes while white pixels correspond to cells.

$$v(i,j) = \begin{cases} (1,0), & \text{if } 0 \leq p(i,j) < 10 \\ (0,0), & \text{if } 10 \leq p(i,j) \leq 245 \\ (0,1), & \text{if } 245 < p(i,j) \leq 255 \end{cases}, \qquad (1)$$

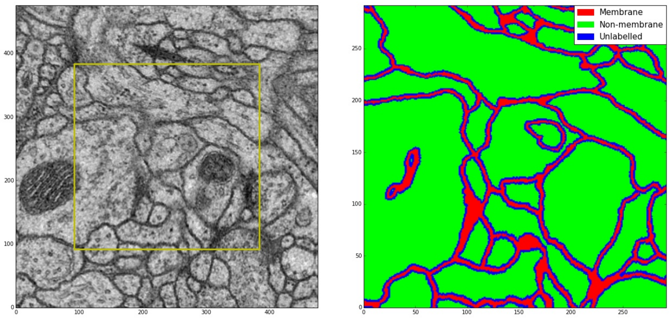
where $(1, 0)$ indicates the pixel belonging to the 'membrane' class and $(0, 1)$ the 'cell' class, while $(0, 0)$ was considered as 'unlabeled', with i and j specifying the location of the pixel in the image. Pixels with a $(0, 0)$ label vector have zero contribution to the cross entropy loss function. An example input patch and the resulting labeling is given in Figure 2. The yellow square in Figure 2a indicates the output region of the network, which corresponds to the pixel labels in Figure 2b, where green indicates cells, red the membranes and blue unlabeled pixels.

IV. Model Architecture

The U-net architecture from Ronneberger et al. [4], depicted in Figure 3, consists of a downsampling pathway followed by an upscaling pathway. Each corresponding level in the two pathways is connected by a number of skip-connections, each acting as a channel over which higher

[1] Weakly labeled data typically comprises labels for complete images or bounding boxes around regions, while strongly labeled data refers to pixel-level segmentation maps.

[2] An IPython notebook outlining how the transformations were applied is available at http://cs.sun.ac.za/~kroon/docs/EMTransformations.zip



(a) Input patch

(b) Label patch

Fig. 2. An example input patch (2a) where the region in the yellow square corresponds to the pixel labels in 2b, where green indicates cells, red the membranes and blue the unlabeled pixels.

resolution information (which might otherwise be lost during downsampling) can be transferred. All convolutional layers perform valid convolutions using 3×3 filters and have rectified linear unit (ReLU) activation functions. The downsampling path contains multiple maxpooling layers, each performing a 2×2 pooling operation with a stride of 2. In contrast, the upscaling pathway uses multiple upsampling layers, which each perform an upsampling of the feature maps followed by a 2×2 convolution to halve the number of feature maps. The output layer then performs a 1×1 convolution to produce the same number of feature maps as the number of classes in the data set, before applying softmax normalization over these feature maps to produce a probability distribution for each pixel in the image. Dropout [16] was also applied on the incoming and outgoing connections of the deepest level. The resulting model also utilized a custom pixel weighting function, which allowed state-of-the-art performance with a Rand Score Thin metric (see Section VI) value of 97.27. It is worth noting that the U-net architecture was reimplemented in our system, based on the architectural information that was made available (not including the pixel weighting function) and our implementation yielded a score of 95.94 as the best out of three submissions.

Some changes were made to the original architecture to simplify and generalize it. Firstly, the dropout layers were removed as they were deemed unnecessary: Ronneberger et al. [4] motivated the use of dropout as a form of data augmentation; however, given the amount of data enrichment that we performed (set out in Section III), initial experiments indicated that the dropout layers had no significant effect on the performance. Secondly, the upsampling layers were replaced with deconvolutional layers, performing backwards strided convolution [3]. The deconvolutional layers had a filter size of 5×5 with a stride of 2. This allowed the upscaling and halving of the number of feature maps to be done in a single, trainable operation. Lastly, the ReLU activation functions were replaced with their recently introduced more general form, parameterized ReLUs (PReLU), since this has led to improved performance on other image-based tasks [17], [18].

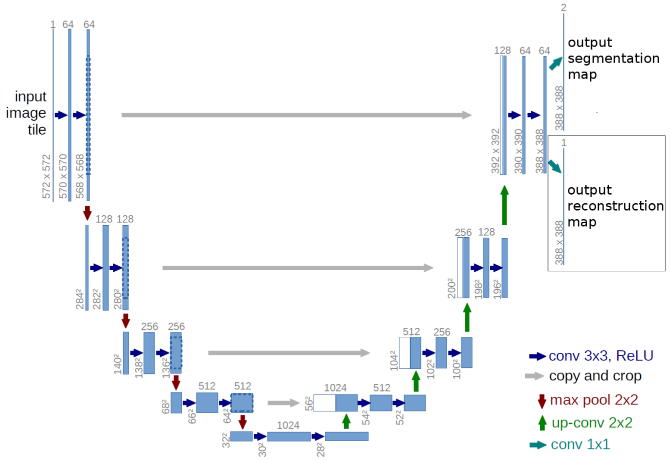


Fig. 3. The U-net architecture [4]. The box on the right hand side indicates an additional reconstruction layer not present in the original network, but added in this study.

To accommodate the use of unsupervised pre-training on this network, we augmented the network with an extra output layer parallel to the softmax output layer — see the boxed region in Figure 3. Similar to the softmax output layer, the additional output layer (henceforth referred to as the reconstruction layer) performed a 1×1 convolution and used a linear activation function. This allowed unsupervised learning to be performed in an end-to-end fashion, by requiring the network to reconstruct the received input image. This is different from the standard greedy layer-wise approach, where each layer in the network is consecutively trained to reconstruct its own input. Having the softmax output layer and the reconstruction layer in parallel allows supervised and unsupervised training to be performed in a single training session, using an extra control parameter in the cost function to switch smoothly between these two modes of training, as discussed in more detail below.

V. Training

The network architecture was implemented using Theano [19], [20] in Python and models were trained on a desktop workstation containing an Intel Core i7-4790 3.6GHz CPU, 16GB of main memory and an NVIDIA GeForce GTX980 Ti graphics processor with 6GB of memory.

Following the sampling technique set out in Section III, a batch of 30 training images were generated per epoch and iterated through one image at a time. Similar to Ronneberger et al. [4], one large image per mini-batch was favored over multiple smaller images, resulting to an input patch of size 476×476 and output size of 292×292. Training was done over 200 epochs using ADADELTA [21], resulting in a training time of about 3 hours per experiment.

As mentioned in Section IV, both supervised and unsupervised learning can be performed in parallel given an extra

TABLE I
RAND SCORE THIN RESULTS FOR ALL 20 EXPERIMENTS.

No.	Supervised Model	Pre-trained Model
1	96.17	97.53
2	97.86	97.69
3	97.12	97.96
4	96.90	97.00
5	97.88	97.25
6	97.67	97.55
7	98.39	97.76
8	98.05	98.07
9	97.46	97.50
10	96.83	97.42
Mean ± SD	97.433 ± 0.673	97.573 ± 0.317

control parameter. As such, the cost function to be minimized is given by

$$E = \beta(t) L_S + (1 - \beta(t)) L_R, \tag{2}$$

where L_S is the softmax loss (standard cross entropy loss averaged over all pixels), L_R is the reconstruction loss (standard mean squared error) and $0 \leq \beta(t) \leq 1$ encodes the tradeoff between these two loss functions. In our experiments, we set $\beta(t)$ to the shifted sigmoid

$$\beta(t) = \frac{1}{1 + \exp(K - t)}, \tag{3}$$

where t is the current epoch number and K is a parameter which can roughly be seen as the epoch number at which the transition should occur. For our experiments, $K = 50$ was found to be sufficient to ensure pre-training convergence.

This choice of $\beta(t)$ ensured a smooth transition to focus primarily on unsupervised learning at the start of training and supervised learning at the end of training. In the purely supervised case, $\beta(t)$ was simply set to one for all epochs.

VI. EXPERIMENTS AND RESULTS

A total of 20 experiments were performed following the procedures set out in Sections III–V, 10 for the purely supervised case and 10 for the pre-trained case. The same labeled training data was used both with and without unsupervised pre-training to facilitate comparing the two scenarios. We generated 10 random numbers which were used as seeds in both cases, hence the only difference between the two was the additional cost L_R from the reconstruction layer. All models were evaluated on the test set using the Fiji script [22] provided by the organizers of the ISBI challenge. Currently, the script reports two metrics: foreground-restricted Rand scoring after border thinning (Rand Score Thin) and foreground-restricted information-theoretic scoring after border thinning (Information Score Thin). These metrics are quite complex—full details are available in Arganda-Carreras et. al. [23]. The cited paper also notes that the Rand Score Thin metric is considered more robust; thus we use it for all our experiments.

The metric value outputs for both cases, as well as the resulting means and standard deviations are shown in Table I. Figure 4 presents a boxplot illustrating the differences between the result distributions for the two approaches. We also provide the output of each approach on a few example images for qualitative comparison in Figure 5.

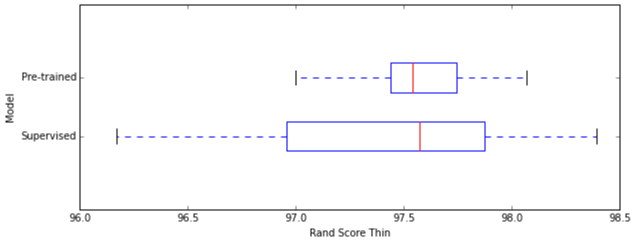


Fig. 4. Boxplot of the distributions for the purely supervised and pre-trained models in Table I ($n = 10$ in each case). Higher values are better.

TABLE II
THE P-VALUES FOR VARIOUS STATISTICAL TESTS

Mean Tests	
t-test	Mann-Whitney U test
0.56822	0.73373

Variance Tests			
F-test	Levene's Test	Bartlett's Test	Brown-Forsythe Test
0.00943	0.02872	0.03553	0.04246

The values in Table I were then used in a battery of hypothesis tests, testing for both equality of means and equality of variance of the two distributions at the 5% significant level. Since we did not expect the results to be normally distributed, we mainly investigated tests that did not require it. We do, however, provide the classical test results assuming normality for comparison. The t-test and Mann-Whitney U-test [24] were used to test for equality of means, while the F-test, Levene's test [25], Bartlett's test [26] and the Brown-Forsythe test [27] were used to test equality of variances. The p-values corresponding to the different tests are shown in Table II.

VII. DISCUSSION

The metric used, Rand Score Thin, is calculated at different thresholds, after which the best result is reported. Arganda-Carreras et al. [23] identifies 4 types of errors that the metric is sensitive to, namely the splitting of cells, the merging of cells and the complete addition or removal of cells. Even with this information, it is still challenging to qualitatively distinguish which approach performs better. There are some differences that are immediately apparent between the two, as pointed out by the green boxes in Figure 5, but in no way is it indicative of which approach is better, which leaves only a quantitative comparison.

As clearly shown in Figure 4, the metric results in Table I indicate that the pre-trained model performed slightly better on average and had a tighter (lower variance) output distribution. The t-test and Mann-Whitney U test both failed to reject the hypothesis of equal mean scores for the two approaches. Thus, our experiments were not sufficient to detect any possible underlying difference in the average performance of the models. Previous work suggests that models trained in a true semi-supervised setting (with few labeled data), can show improved

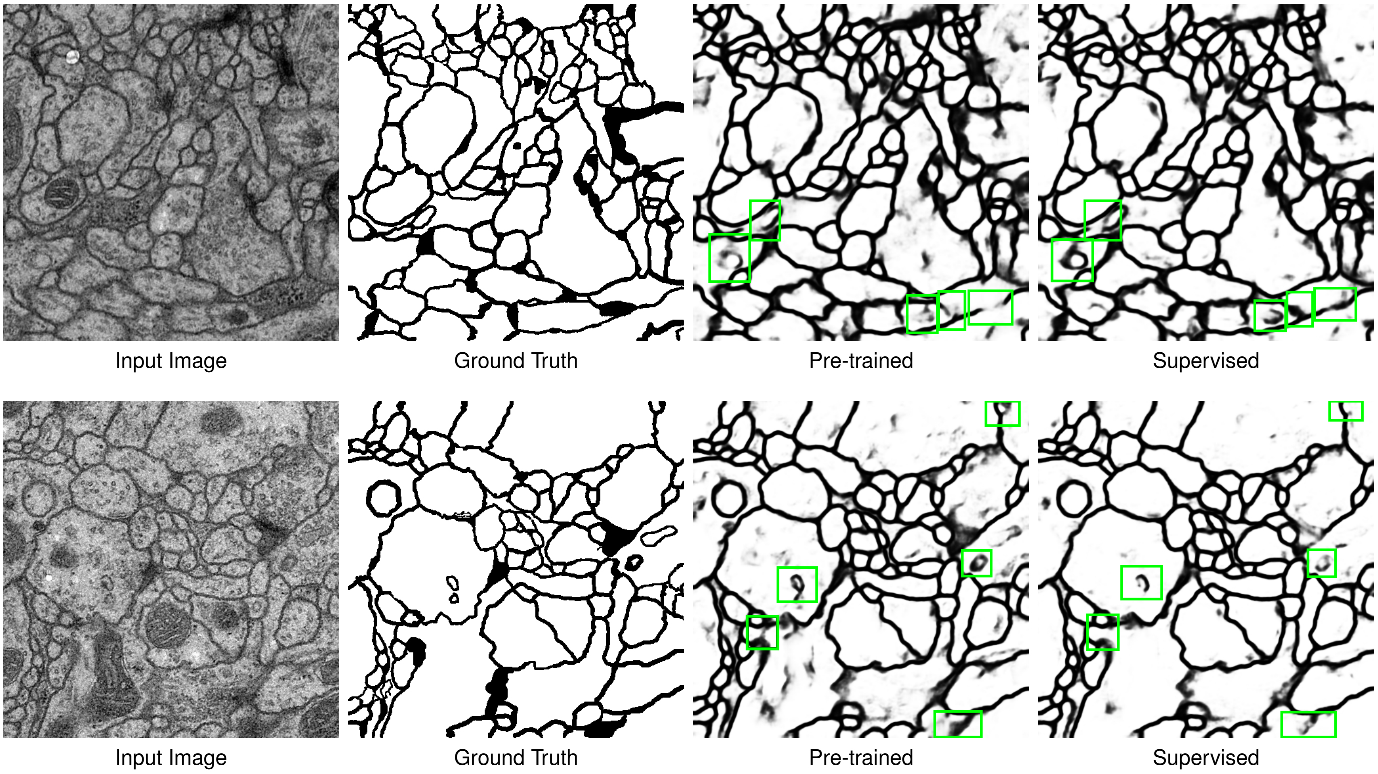


Fig. 5. Two examples from the test set showing the segmentation output of the pre-trained model compared to the purely supervised model. Highlighted in the green boxes were some of the most apparent differences that could potentially have an influence on the metric value.

performance over its purely supervised counterpart [1], [10], [11], [12]. However, our failure to obtain such an improvement still makes sense in this setting, due to the fact that all data used for training was labeled. Additional unlabeled data provides information about the expected distribution of inputs, allowing the classifier to focus more effectively on relevant portions of the input space when the labeled training data does not adequately represent the input distribution. In our setting, since no additional unlabeled data is provided, our classifier could not benefit from this.

The remaining statistical tests, the F-test, Levene's test, Bartlett's test and the Brown-Forsythe test, evaluated the hypothesis of equal variances of the metric under both approaches. All these tests rejected the null hypothesis at a 5% significance level, suggesting that it is improbable that the difference in observed variance for the two approaches was by chance, despite the small sample size. Given that the only difference between the experiments for each case was their initialization, the reduction in the variance for the pre-trained model suggests that unsupervised pre-training via the reconstruction loss made the model more robust to random initialization. This result aligns well with the findings of Erhan et al. [1], who make a case (for much smaller networks) that unsupervised pre-training acts as a regularizer which adds robustness to random initialization and as such, reduces the variance in the model performance.

Converting a purely supervised FCN to one capable of undergoing unsupervised pre-training is fairly straightforward, provided that the output of the FCN is of the same scale as the original input, i.e. the FCN is used to produce a semantic segmentation map of the input. The particular architecture of U-net, specifically the presence of the skip-connections, did pose an interesting challenge. The skip-connections could potentially act as short-cuts during unsupervised training, leading to little or no benefit for the deeper levels. Upon further investigation, this was indeed the case. Good reconstruction after training was entirely dependent on the top-most set of skip-connections — this was determined by setting the weights of the individual skip-connections on the various levels in the architecture to zero and measuring the difference in performance. This suggests more work is needed in augmenting an FCN with skip-connections to allow unsupervised pre-training that is beneficial to the entire network, not just a small portion of it. One such approach would be to have multiple reconstruction layers, one on each level in the architecture, with the objective of reconstructing the input for the respective level it is attached to.

VIII. Conclusion

We proposed a novel augmentation for FCNs which allows end-to-end unsupervised learning to be used as a pre-training step. Analysis suggested that performing unsupervised pre-training provides a statistically significant reduction in the variance of the model performance compared to a purely supervised FCN. This reduction in variance further supports the generalizer hypothesis of Erhan et al. [1], which suggests that unsupervised pre-training adds robustness to the model against random initialization, reducing the model variance accordingly. Lastly, we observed that the skip-connections in the U-net architecture allowed unsupervised learning to bypass the deeper levels of the network, suggesting that a more robust approach is needed to reap the full benefits of unsupervised learning.

In future work, we plan on finding an approach such that the entire FCN can benefit from unsupervised learning, while maintaining the end-to-end training aspect of the model. This also includes exploring different cost functions to integrate the reconstruction cost with the supervised classification cost. The ability to perform unsupervised pre-training also opens up the potential to add a denoising component to the reconstruction task by requiring the network to reconstruct original images from corrupted inputs — this is analogous to the use of denoising autoencoders [28]. Lastly, we plan on redoing the pre-training experiments with extra unlabeled data from the original test set, bringing it closer to true semi-supervised conditions and to compare with the results we have already obtained.

References

[1] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756025

[2] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[5] H. Chen, X. Qi, J.-Z. Cheng, and P.-A. Heng, "Deep Contextual Networks for Neuronal Structure Segmentation." [Online]. Available: http://appsrv.cse.cuhk.edu.hk/~hchen/research/2012isbi_seg.html

[6] S. Wiehman and H. de Villiers, "Semantic segmentation of bioimages using convolutional neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, 2016.

[7] R. P. Poudel, P. Lamata, and G. Montana, "Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation," *arXiv preprint arXiv:1608.03974*, 2016. [Online]. Available: http://arxiv.org/abs/1608.03974

[8] H. Chen, Q. Dou, X. Wang, J. Qin, J. C. Y. Cheng, and P.-A. Heng, *3D Fully Convolutional Networks for Intervertebral Disc Localization and Segmentation*. Cham: Springer International Publishing, 2016, pp. 375–382. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-43775-0_34

[9] (2015) Miccai 2015 challenge: Automatic intervertebral disc (ivd) localization and segmentation from 3d t2 mri data. [Online]. Available: http://ijoint.istb.unibe.ch/challenge/index.html

[10] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Advances in Neural Information Processing Systems*, 2015, pp. 1495–1503.

[11] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3581–3589. [Online]. Available: http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf

[12] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with Ladder networks," in *Advances in Neural Information Processing Systems*, 2015.

[13] H. Valpola, "From neural PCA to deep unsupervised learning," *Adv. in Independent Component Analysis and Learning Machines*, pp. 143–171, 2015.

[14] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein, "An integrated micro- and macroarchitectural analysis of the *Drosophila* brain by computer-assisted serial section electron microscopy," *PLoS Biol*, vol. 8, no. 10, pp. 1–17, 10 2010. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pbio.1000502

[15] A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tomancak, V. Hartenstein, and R. J. Douglas, "Trakem2 software for neural circuit reconstruction," *PLoS ONE*, vol. 7, no. 6, pp. 1–8, 06 2012. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0038011

[16] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[18] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[19] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.

[20] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[21] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[22] I. Arganda-Carreras. (2016) Segmentation evaluation after border thinning - script. [Online]. Available: http://imagej.net/Segmentation_evaluation_after_border_thinning_-_Script

[23] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, T. Liu, M. Seyedhosseini, T. Tasdizen, L. Kamentsky, R. Burget, V. Uher, X. Tan, C. Sun, T. D. Pham, E. Bas, M. G. Uzunbas, A. Cardona, J. Schindelin, and H. S. Seung, "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Frontiers in Neuroanatomy*, vol. 9, p. 142, 2015. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnana.2015.00142

[24] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, 03 1947. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177730491

[25] H. Levene, "Robust tests for equality of variances," *Contributions to probability and statistics: Essays in honor of Harold Hotelling*, vol. 2, pp. 278–292, 1960.

[26] G. W. Snedecor and W. G. Cochran, "Statistical methods, 8th edition," *Ames: Iowa State Univ. Press Iowa*, 1989.

[27] M. B. Brown and A. B. Forsythe, "Robust tests for the equality of variances," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 364–367, 1974.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.