Enhanced Methods for Lymphocyte Detection and Segmentation on H&E Stained Images using eXclusive Autoencoders

Chao-Hui Huang, Daniel Racoceanu

Enhanced Methods for Lymphocyte Detection and Segmentation on H&E Stained Images using eXclusive Autoencoders

Chao-Hui HUANG[1] and Daniel RACOCEANU[2]

***Abstract*— In this paper, we propose a generalized solution for lymphocyte detection and segmentation, based on a novel image feature extraction method, named exclusive autoencoder (XAE). XAE is compatible with conventional autoencoder (AE) and able to provide additional information about the categorization in the feature space. For the task of lymphocyte detection, XAE was able to reach the an F-score of 99.96%, outperforming the state-of-the-art methods (reporting an F-score of 90% [1]). Further, based on the integration of XAE+FCN (fully connected network) and conventional image processing function blocks provided in CellProfiler, we propose a lymphocyte segmentation pipeline. The obtained Dice coefficient reached 88.31% while the cutting-edge approach was at 74% [2].**

I. INTRODUCTION

Image processing at the scale of nucleus is a critical step for computer-aided pathology [3]. Although immuno-histo-chemical (IHC) antibody staining produces images for more specific purposes, the conventional hematoxylin & eosin (H&E) staining remains the golden standard for routine cancer diagnosis.

Automated lymphocyte (as well as the general nuclei) detection, classification and segmentation problems are important in clinical studies and have attracted a huge number of researchers' attention, *e.g.*, for detection problems, Xu *et al.* [4] proposed methods of stacked sparse autoencoder (SSAE); Khoshdeli *et al.* [5] reported convolutional neural network (CNN) based approaches. For nuclei detection and classification, Sirinukunwattana *et al.* proposed a locality sensitive deep learning method [6] which took the advantages of location sensitivity on the given images. Further, Janowczyk *et al.* introduced a deep-learning-based lymphocyte classification [1]. For lymphocyte segmentation, most of the existing approaches were based on traditional computer vision algorithms or conventional machine learning methods. Recently, deep learning took also the lead regarding this challenging task [7].

An autoencoder (AE) is a key component in many deep learning (DL) based approaches (*e.g.*, [4], [6]). However, as a unsupervised neural network (NN), a conventional AE has its limitations, including: 1) AE does not provide clear clues indicating to which class belongs the obtained feature; 2) a feature obtained using a conventional AE can exist in two or more classes of the given dataset, thus, the feature will not be able to effectively contribute to the classification task; and 3) in a unbalanced dataset, the feature space may be occupied by the majority of the dataset, and, as a result, subtle (but critical) features may be ignored.

In order to palliate these issues, we propose an improved version of AE, called exclusive autoencoder (XAE), which aims at learning the features in the manner which is not only focusing on the common features across classes, but also the exclusive features of each class of the given dataset. Based on this concept, in this paper, we propose a two-step lymphocyte detection approach based on XAE+FCN. The first step performs a coarse detection, which multiple regions of interest as the candidates for further classification. In the second step, we performe true-lymphocyte v.s. false-lymphocyte classification. As a result, a high accurate lymphocyte detection can be achieved.

In addition, we develop a method of lymphocyte segmentation, based on the proposed XAE+FCN architecture. The process steps includes: 1) lymphocyte detection; 2) lymphocyte classification; and 3) the integration with general image segmentation procedure provided in CellProfiler [8]. In this experiment, a lymphocyte H&E image dataset [1] was obtained from the online supplemental materials of the article [1]. The details of these steps will be described in the following sub-sections.

In this paper, we will discuss how does the proposed XAE largely improve the existing AE-based nuclei detection/classification. In the following sections, we will, first, introduce the approach of XAE (Sect. II). Then, in the result section, based on the proposed approaches, the examples of the use of XAE on nuclei detection, classification, and a real world application of lymphocyte segmentation will be presented (Sect. III). Finally, the conclusions will be drawn (Sect. IV).

II. METHOD

The core of the proposed methods of lymphocyte detection segmentation is a combination of XAE and a fully connected network (FCN). XAE, in fact, is an extension of conventional AE, which is a type of neural network aiming at reconstructing the input by a smaller number of hidden units. In other words, AE is able to extract the key components of the given training patterns. This capability has been widely used as a pre-training step of feature extraction. *E.g.,* Sparse AE [9], the output of the hidden layer is sparse, and each unit in the hidden represents a unique feature of the given training patterns. The performance of the following classification

[1]Chao-Hui HUANG is with Pfizer Inc. huangch.tw@gmail.com

[2]Daniel RACOCEANU is with the Department of Engineering, Faculty of Science & Engineering, Sorbonne University daniel.racoceanu@sorbonne-universite.fr

XAE package can be downloaded from: https://github.com/huangch/xae/

[1]http://www.andrewjanowczyk.com/deep-learning/

depends on a probability that one or more hidden units are correlated to the data labels. If this probability can be controlled, the accuracy of the following classification can so be improved.

Given a dataset of two classes, one idea is to partition the hidden layer of AE into two segments, so that given the training patterns of the first data class, the first segment is highly correlated to the first segment, but uncorrelated to the second segment, and *vice visa*. More precisely, given the training patterns of the first data class, we wish the units of the first segment to represent all critical features and the units of the second segment to all be 0. However, this is a very strict requirement. Thus, we pursuit the next best option, that is, making the first segment sparse and the second segment falling into a narrow zero-mean Gaussian distribution. To achieve this goal, we introduce XAE.

The conceptual idea of XAE is to partition the hidden layer into two or more segments so that a segment is only correlated to relevant dataset, according to the given labels. There are two types of hidden layer segments: 1) the segments for exclusive components , which represents the feature representation for a specific class of the dataset; and 2) the segments for mutual components , representing the mutual features across two or more data classes. The XAE configuration can be expanded for datasets with two or more data categories. In the following sections, the further details of XAE architecture will be discussed.

Given a dataset with two or more classes, the labels of the classes can be described as a finite set, $\mathbf{L} = \left\{l_1, l_2, \cdots, l_i, \cdots, l_{|\mathbf{L}|}\right\}$, where each l_i represents a label in the problem. The cardinality, $|\mathbf{L}|$, is the total number of label types in the problem. According to the situation, a dataset may or may not contain data instances corresponding to multiple labels [10]. Thus, instead of using a single label l_i for indicating the category of a dataset, we can use a set of labels, $\mathbf{S}_i$, to describe the involved labels of the i^{th} group of data instances, which share the same label sets, as following: $\mathbf{S}_i \subseteq \mathbf{L}, \mathbf{S}_i \neq \emptyset$. Thus, given a dataset, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k, \cdots\}$, each $\mathbf{x}_k \in \mathbb{R}^M$ is a M dimensional data vector. The subset of all labels, $\mathbf{S}_i$, can be used to indicate the categories of a sub-dataset, $\mathbf{X}^{(\mathbf{S}_i)} \subseteq \mathbf{X}$, representing in the form of a block matrix:

$$\mathbf{X}^{(\mathbf{S}_i)} = \left[\mathbf{x}_1^{(\mathbf{S}_i)}, \mathbf{x}_2^{(\mathbf{S}_i)}, \cdots, \mathbf{x}_k^{(\mathbf{S}_i)}, \cdots\right] \in \mathbb{R}^{M \times |\mathbf{X}^{(\mathbf{S}_i)}|}, \quad (1)$$

where each $\mathbf{x}_k^{(\mathbf{S}_i)}$ represents a data instance corresponding to one or more data categories, according to the defined $\mathbf{S}_i$. In other words, $\mathbf{S}_i$ represents the *priori knowledge* of $\mathbf{X}^{(\mathbf{S}_i)}$.

A. Architecture of XAE

a) Encoder:: Given the input $\mathbf{x}_k^{(\mathbf{S}_i)}$, the hidden layer is obtained by:

$$\mathbf{z}_k^{(\mathbf{S}_i)} = a_{\text{encoder}}\left(\mathbf{W}^T \mathbf{x}_k^{(\mathbf{S}_i)} + \mathbf{b}_{\text{encoder}}\right), \quad (2)$$

where $a_{\text{encoder}}(\cdot)$ represents an activation function for the encoder; $\mathbf{b}_{\text{encoder}} \in \mathbb{R}^N$ is the bias term for the encoder; $\mathbf{z}_k^{(\mathbf{S}_i)}$ is the output of the hidden layer; and $\mathbf{W}$ is the feature set representing in the form of a block matrix, which, in fact, is a concatenation of some subset of all features, *e.g.*,

$$\mathbf{W} = \left[\mathbf{W}_{(M \times n_1)}^{(\mathbf{T}_1)} \ \mathbf{W}_{(M \times n_2)}^{(\mathbf{T}_2)} \ \cdots\right] \in \mathbb{R}^{M \times N} \quad (3)$$

where $N = \sum_j n_j$. Each $\mathbf{T}_j \subseteq \mathbf{L}$, which is similarly defined as of $\mathbf{S}_i$. However, $\mathbf{T}_j$ and $\mathbf{S}_i$ are serving different purposes. That is, each $\mathbf{S}_i$ is used as a meta-variable, mapping $\mathbf{X}^{(\mathbf{S}_i)}$ to one or more labels, while each $\mathbf{T}_j$ represents the desired *segmentation* of $\mathbf{W}^{(\mathbf{T}_j)}$, correlating each $\mathbf{W}^{(\mathbf{T}_j)}$ to one or more data classes.

By following this concept, given a set of $\mathbf{W}^{(\mathbf{T}_j)}$, $\mathbf{z}_k^{(\mathbf{S}_i)}$ coresponds to a concatenation:

$$\mathbf{z}_k^{(\mathbf{S}_i)} = \left[\left(\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_1)}\right)^T \ \left(\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_2)}\right)^T \ \cdots\right]^T \in \mathbb{R}^N, \quad (4)$$

where each

$$\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_j)} = \left[z_{k,1}^{(\mathbf{S}_i, \mathbf{T}_j)}, z_{k,2}^{(\mathbf{S}_i, \mathbf{T}_j)}, \cdots, z_{k,n_j}^{(\mathbf{S}_i, \mathbf{T}_j)}\right]^T \in \mathbb{R}^{n_j}, \quad (5)$$

represents the sub-activation-vector of the given $\mathbf{x}_k^{(\mathbf{S}_i)}$ and its value is related to the relationship between subset of all labels $\mathbf{S}_i$ and $\mathbf{T}_j$. Note that the superscription, $\mathbf{S}_i$, doesn't imply an additional index of $\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_j)}$. Instead, it represents how does the $\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_j)}$ being treated based on the relationship between $\mathbf{S}_i$ and $\mathbf{T}_j$.

b) Decoder:: The decoder performs a partially connected feed-forward neural network [11], in which, the neurons are connected differently based on their input types, *e.g.*,

$$\mathbf{y}_k^{(\mathbf{S}_i)} = a_{\text{decoder}}\left(\sum_{j=1}^{Q^{(\mathbf{S}_i)}} \mathbf{W}^{(\mathbf{T}_j)} \mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_j)} + \mathbf{b}_{\text{decoder}}\right), \quad (6)$$

where $\{\mathbf{T}_j | \mathbf{S}_i \cap \mathbf{T}_j \not\equiv \emptyset, \forall j\}_{Q^{(\mathbf{S}_i)}}$. $a_{\text{decoder}(\cdot)}$ is the activation of the decoder; $\mathbf{W}^{(\mathbf{T}_j)}$ is a subset of all features defined in Eq. (3); $\mathbf{b}_{\text{decoder}} \in \mathbb{R}^M$ is the bias term of the decoder; $\mathbf{y}_k^{(\mathbf{S}_i)}$ is the output of the decoder, which is reconstruction of the $\mathbf{x}_k^{(\mathbf{S}_i)}$ of the encoder defined in Eq. (2); $Q^{(\mathbf{S}_i)} = |\{\mathbf{T}_j | \mathbf{S}_i \cap \mathbf{T}_j \not\equiv \emptyset, \forall j\}|$ is the number of all $\mathbf{T}_j$ satisfying this condition with the given $\mathbf{S}_i$. In other words, $\mathbf{y}_k^{(\mathbf{S}_i)}$ is computed using a subset of all possible $\mathbf{T}_j$, each of which is a *non-disjoint set* of $\mathbf{S}_i$.

The cost function of an XAE is defined as:

$$\begin{aligned}
\mathcal{J}_{\text{XAE}} = & \frac{1}{2|\mathbf{X}^{(\mathbf{S}_i)}|} \sum_{i,k} ||\mathbf{x}_k^{(\mathbf{S}_i)} - \mathbf{y}_k^{(\mathbf{S}_i)}||_2^2 + \frac{\lambda}{2} ||\mathbf{W}||_F^2 \\
& + \alpha v_{\mathbf{S}_i, \mathbf{T}_j} \sum_{i,j} \left(\frac{1}{n_j} \sum_{m=1}^{n_j} \text{KL}\left(\rho || \rho_m^{(\mathbf{S}_i, \mathbf{T}_j)}\right)\right) \\
& + \frac{\beta \bar{v}_{\mathbf{S}_i, \mathbf{T}_j}}{2} \sum_{i,j} \left(\frac{1}{n_j |\mathbf{X}^{(\mathbf{S}_i)}|} \sum_{k=1}^{|\mathbf{X}^{(\mathbf{S}_i)}|} \sum_{m=1}^{n_j} H_a\left(\tau_{0,\sigma} || z_{k,m}^{(\mathbf{S}_i, \mathbf{T}_j)}\right)\right)^2 \\
& + \frac{\gamma \bar{v}_{\mathbf{S}_i, \mathbf{T}_j}}{2} \sum_{i,j} \left(\frac{1}{|\mathbf{X}^{(\mathbf{S}_i)}|} \sum_{k=1}^{|\mathbf{X}^{(\mathbf{S}_i)}|} \left(\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_j)}\right) \left(\mathbf{z}_k^{(\mathbf{S}_i, \mathbf{T}_j)}\right)^T - b\mathbf{I}\right)^2,
\end{aligned} \quad (7)$$

where α, β and γ are parameters controlling the weights of sparseness and exclusiveness;

$$\rho_m^{(\mathbf{S}_i,\mathbf{T}_j)} = \frac{1}{|\mathbf{X}^{(\mathbf{S}_i)}|}\sum_{k=1}^{|\mathbf{X}^{(\mathbf{S}_i)}|} z_{k,m}^{(\mathbf{S}_i,\mathbf{T}_j)} \tag{8}$$

is the sparsity measurement of the m^{th} element of the sub-activation-vector $\mathbf{z}_k^{(\mathbf{S}_i,\mathbf{T}_j)}$ in Eq. (5); $\mathrm{KL}(\rho||\cdot)$ is the measurement of Kullback–Leibler (KL) divergence, which requires a parameter ρ, indicating the level of sparsity; and

$$H_a\left(\tau_{0,\sigma}||z_{k,m}^{(\mathbf{S}_i,\mathbf{T}_j)}\right) = G_a\left(z_{k,m}^{(\mathbf{S}_i,\mathbf{T}_j)}\right) - G_a(\tau_{0,\sigma}) \tag{9}$$

measures the Gaussianity of the given $z_{k,m}^{(\mathbf{S}_i,\mathbf{T}_j)}$ so that the response of an excluded feature approximates to a Gaussian random variable defined by $\tau_{0,\sigma}$, which is a zero-mean Gaussian random variable with a small standard deviation of σ. Finally, $\upsilon_{\mathbf{S}_i,\mathbf{T}_j}$ represents a semaphore determined by the relationship between $\mathbf{S}_i$ and $\mathbf{T}_j$:

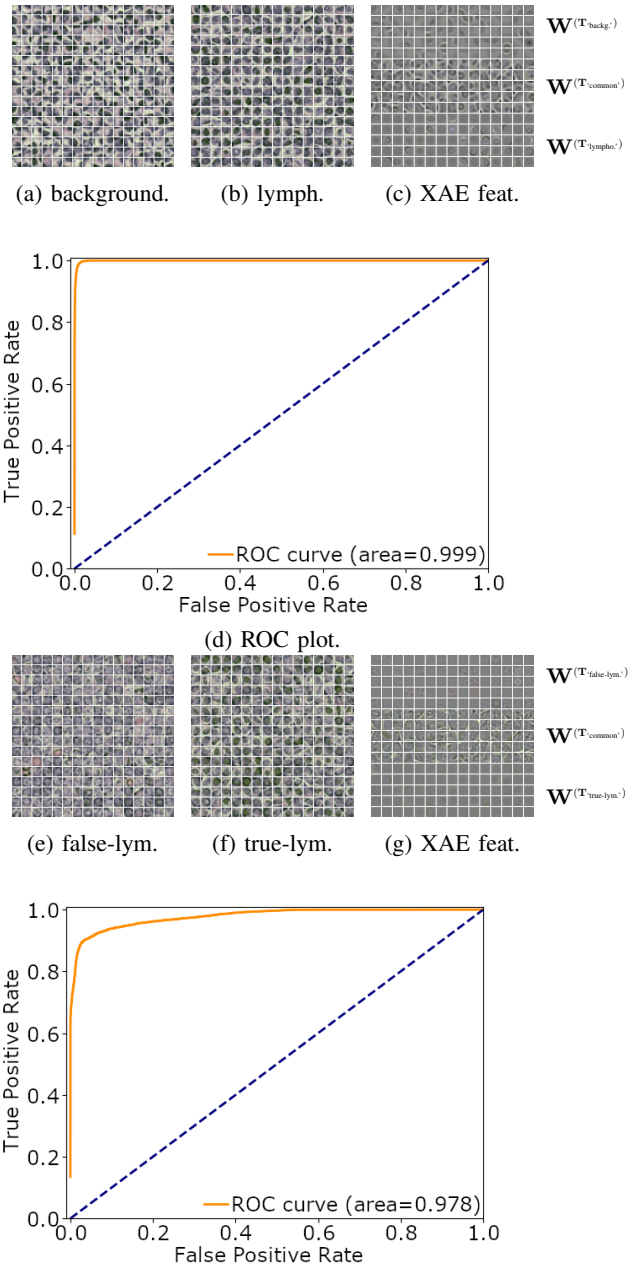
$$\upsilon_{\mathbf{S}_i,\mathbf{T}_j} = \begin{cases} 1, \text{ if } \mathbf{S}_i \cap \mathbf{T}_j \not\equiv \emptyset \\ 0, \text{ otherwise} \end{cases}, \bar{\upsilon}_{\mathbf{S}_i,\mathbf{T}_j} = 1 - \upsilon_{\mathbf{S}_i,\mathbf{T}_j}. \tag{10}$$

III. Results

In lymphocyte detection, first, we cropped lymphocyte images according to the annotations provided by the dataset. Since the negative (image background) samples were not provided in the dataset, we used the same approach proposed by Janowczyk *et al.* [1], that is, cropping image patches with the exclusion locations annotated as lymphocyte based on a Baysan's naïve classifier. Then, we performed XAE+FCN for the lymphocyte / background classification. Some image patterns can be found in Fig. 1a and 1b; the corresponding XAE features are showing in Fig. 1c; the ROC plot can be found in Fig. 1d At this step, the F-score was 98.67%. Meanwhile, for comparison, we also tested AE+FCN and the obtained F-score was 98.29%.

Although at this stage, we obtained a good accuracy, a major issue was that most of the false-positives were other cell types (see Fig. 1e). As a result, the performance of lymphocyte detection was impacted. To tackle this problem, we introduced an additional step of true-lymphocyte v.s. false-lymphocyte classification (here, the false-lymphocytes mainly represent image patterns of other nucleus-like structures). The training patterns for true-lymphocytes were the same as in the previous step. The false-lymphocytes were the false-positives from the results of the previous step (see Fig. 1e and 1f). The corresponding XAE features are shown in Fig. 1g and the receiver operating characteristic (ROC) plot is shown in Fig. 1h.

By integrating the above steps, we obtained the overall F-score at 99.96% for lymphocyte detection. The performance can be compared with the cutting-edge technology reported in [1], in which, Janowczyk *et al.* obtained an F-score of 90%. In addition, we also compared this performance with the one using AE+FCN. In this case, the obtained F-score was 99.38%. The performance was behind the proposed XAE+FCN due to the fact that in the second step, XAE+FCN



(a) background. (b) lymph. (c) XAE feat.

(d) ROC plot.

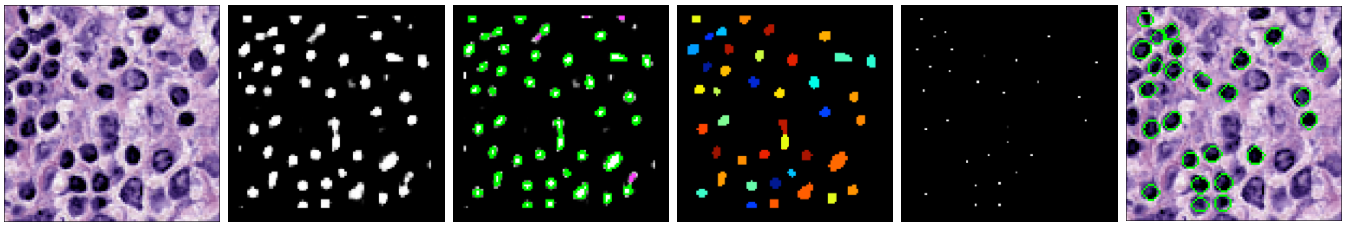
(e) false-lym. (f) true-lym. (g) XAE feat.

(h) ROC plot.

Fig. 1: The corresponding images of lymphocyte detection/classification and the corresponding ROC plots: (a)-(d) lymphocyte detection; (e)-(h) lymphocyte classification.

outperformed AE+FCN. Further, we compared the result with cutting edge approach *e.g.*, a locality sensitive deep learning approach [6] proposed by Sirinukunwattana *et al.*. In their report, the authors reported a performance of nearly 80% for inflammatory categories, while the outcome of XAE+FCN in a similar configuration was corresponding to an F-score of 92.39%. In order to also compare with the conventional AE, the obtained F-score of AE+FCN was 77.13%.

Finally, we built a pipeline based on CellProfiler [8], with the integration of the proposed XAE+FCN approach. The scenario can be found in Fig. 2. We used the data provided in the supplemental materials of [1] as the ground truth. The



(a) (b) (c) (d) (e) (f)

Fig. 2: The procedure of lymphocyte segmentation: (a) the input; (b) a probability map obtained from the proposed lymphocyte detection based on XAE+FCN, where the intensity represents the probability that a pixel belongs to a lymphocyte; (c) performing general object detection on the probability map obtained from step (b); (d) performing general object segmentation on the result of object detection, where each object is either a lymphocyte or a false-positive; (e) the result of lymphocyte classification based on XAE+FCN, where the intensity represents the probability that a pixel corresponds to the center of a lymphocyte; and (f) the final result of lymphocyte segmentation obtained by integrating steps (d) and (e).

Layer	**Dimensions**		**Activate Functions**	
XAE Input Layer	11 (pixel) × 11 (pixel) × 3 (RGB)		N/A	
XAE Hidden Layer	exclusive features for positives	exclusive features for negatives	encoder	decoder
	225	225	sigmoid	linear
FCN Hidden Layer 1	225		lrelu	
FCN Hidden Layer 2	45		lrelu	
FCN Hidden Layer 3	9		lrelu	
Classification Layer	2		softmax with logits	

TABLE I: The network configuration for the FCN used in the experiments, the parameters: $\lambda = 1$, $\beta = 1$, $\gamma = 1$, $\rho = 0.05$; the learning rates $\eta_{\text{XAE}} = 0.00005$, $\eta_{\text{FCN}} = 0.00001$; the batch sizes $B_{\text{XAE}} = 1000$, $B_{\text{FCN}} = 1000$; and the iterations $T_{\text{XAE}} = 1000000$, $T_{\text{FCN}} = 1000000$; The chosen optimizers were Adam [12] and its heirs.

data included the manually annotated contours of a subset of all labeled lymphocytes. The obtained Dice coefficient [13] achieved 88.31%, while the best reported performance based on the same dataset was 74% [2].

IV. Conclusions

In this paper, a novel architecture of autoencoder (AE), named exclusive autoencoder (XAE), was discussed for performing lymphocyte detection and segmentation. An XAE provides a solution where the hidden units can learn the features according to the relationship between the extracted feature set and the corresponding label set of the given dataset. In XAEs, the exclusive hidden units learn features of the relevant classes while the features remain uncorrelated to the irrelevant classes of the dataset. On the other had, the common hidden units learn the mutual features across some (or all) classes of the given dataset.

In our experiments, we evaluated the proposed XAE+FCN for lymphocyte detection and segmentation, and reached the F-score at 99.96% while the cutting-edge technology was at 90%. Also, the proposed segmentation method was able to achieve a Dice coefficient of 88.31% while the cutting-edge best record was at 74% on the same dataset.

In the experiments, we have found that in a simpler classification task (*e.g.* lymphocytes v.s. background classification), XAE didn't show advantages in the comparison with a AE-based solution. However, in a more difficult scenario (*e.g.*, lymphocytes v.s. other nuclei types), XAE helped to improve the performance.

References

[1] A. Janowczyk *et al.*, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases.," *Journal of Pathology Informatics*, vol. 7, no. 1, p. 29, 2016.

[2] M. Kuse *et al.*, "A Classification Scheme for Lymphocyte Segmentation in H&E Stained Histology Images," pp. 235–243, Springer, Berlin, Heidelberg, 2010.

[3] O. Rujuta *et al.*, "Review of Nuclei Detection, Segmentation in Microscopic Images," *Journal of Bioengineering & Biomedical Science*, vol. 07, pp. –, may 2017.

[4] J. Xu *et al.*, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 119–130, jan 2016.

[5] M. Khoshdeli *et al.*, "Detection of Nuclei in H&E Stained Sections Using Convolutional Neural Networks.," *IEEE-EMBS International Conference on Biomedical and Health Informatics. IEEE-EMBS International Conference on Biomedical and Health Informatics*, vol. 2017, pp. 105–108, feb 2017.

[6] K. Sirinukunwattana *et al.*, "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1196–1206, may 2016.

[7] J. Chen *et al.*, "Automatic lymphocyte detection in h&e images with deep neural networks," *ArXiv: Computer Vision and Pattern Recognition (cs.CV)*, dec 2016.

[8] A. E. Carpenter *et al.*, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes," *Genome Biology*, vol. 7, p. R100, oct 2006.

[9] A. Ng, "CS294A lecture notes: sparse autoencoder,"

[10] Z.-H. Zhou *et al.*, "Multi-instance multi-label learning," *Artificial Intelligence*, vol. 176, pp. 2291–2320, jan 2012.

[11] S. Kang *et al.*, "Partially connected feedforward neural networks structured by input types," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, 2005.

[12] D. P. Kingma *et al.*, "Adam: A Method for Stochastic Optimization," dec 2014.

[13] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, pp. 297–302, jul 1945.