Supplementary: Truncated Normal Mixture Prior Based Deep Latent Model for Color Normalization of Histology Images

Suman Mahapatra and Pradipta Maji

SI. Proof of Some Important Results

In this section, the proofs of important proposition, theorem, stated in the main manuscript, are provided in detail.

Proof of Proposition 1

For a fixed combination of $\mathcal{E}_c$, $\mathcal{E}_s$ and $\mathcal{G}$, the learning criterion for discriminator $\mathcal{D}$ is to minimize $J_1(\mathcal{E}_c, \mathcal{E}_s, \mathcal{G}, \mathcal{D})$ with respect to $\mathcal{D}$ by computing partial derivative as follows:

$$\frac{\partial J_1(\mathcal{E}_c, \mathcal{E}_s, \mathcal{G}, \mathcal{D})}{\partial \mathcal{D}[x, z_c, z_s]} = 2P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s)(\mathcal{D}[x, z_c, z_s] - A) + 2P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)(\mathcal{D}[x, z_c, z_s] - B) \quad (1)$$

Let the optimal discriminator be denoted as $\mathcal{D}^*[x, z_c, z_s]$. So, at optimal point corresponding to the discriminator $\mathcal{D}$:

$$\left.\frac{\partial J_1(\mathcal{E}_c, \mathcal{E}_s, \mathcal{G}, \mathcal{D})}{\partial \mathcal{D}[x, z_c, z_s]}\right|_{\mathcal{D}=\mathcal{D}^*} = 0;$$

$$\Rightarrow 2\{P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s)(\mathcal{D}^*[x, z_c, z_s] - A) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)(\mathcal{D}^*[x, z_c, z_s] - B)\} = 0; \quad \text{[using (1)]}$$

$$\Rightarrow \mathcal{D}^*[x, z_c, z_s] = \frac{A.P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + B.P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)}{P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)} \quad (2)$$

Proof of Theorem 1

(**IF**) If $P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}} = P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}$, then (2) can be rewritten as follows:

$$\mathcal{D}^*[x, z_c, z_s] = \frac{A.P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + B.P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)}{P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)}$$

$$= \frac{(A+B)P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s)}{2P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s)} = \frac{(A+B)}{2}. \quad (3)$$

Substituting the pre-assumed label values $A = 1$ and $B = 0$, (3) leads to the following:

$$\mathcal{D}^*[x, z_c, z_s] = \frac{(1+0)}{2} = \frac{1}{2}. \quad (4)$$

(**ONLY IF**) Now, (6) of Section II-C1 of the main manuscript can be simplified as follows:

$$(\mathcal{E}_c, \mathcal{E}_s, \mathcal{G})^*$$

$$= \int_{\{x, z_c, z_s\}} \frac{\left[\begin{array}{c}(C-A)P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + \\ (C-B)P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)\end{array}\right]^2}{\left[P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)\right]} dx dz_c dz_s$$

$$= \int_{\{x, z_c, z_s\}} \frac{\left[\begin{array}{c}(C-A)[P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + \\ P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)] + \\ (A-B)P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)\end{array}\right]^2}{\left[P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)\right]} dx dz_c dz_s \quad (5)$$

The real encoding and the generated/fake encoding are considered to have label values $A = 1$ and $B = 0$, respectively. Now, if $\mathcal{D}^*$ attains a probability value $\frac{1}{2}$, that means the optimal discriminator cannot distinguish between real and generated/fake encoding. As a result, label C is eventually assumed to have a value $\frac{1}{2}$. Hence, (5) can be rewritten as

$$(\mathcal{E}_c, \mathcal{E}_s, \mathcal{G})^*$$

$$= \frac{1}{4} \int_{\{x, z_c, z_s\}} \frac{\left[\begin{array}{c}2P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s) - \\ [P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + \\ P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)]\end{array}\right]^2}{[P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)]} dx dz_c dz_s$$

$$= \frac{1}{4} \chi^2_{Pearson} \left[\begin{array}{l}2P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s) \;||\; (P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) \\ \qquad\qquad + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s))\end{array}\right], \quad (6)$$

where $\chi^2_{Pearson}(.)$ represents the Pearson's chi-squared divergence measure. Pearson's chi-squared divergence is a special case of f-divergence measure. The f-divergence between two continuous probability distributions $\mathcal{M}$ and $\mathcal{N}$ is defined as:

$$D_f(\mathcal{M} \,||\, \mathcal{N}) = \int_x \mathcal{N}(x) f\left(\frac{\mathcal{M}(x)}{\mathcal{N}(x)}\right) dx = E_{\mathcal{N}}\left[f\left(\frac{\mathcal{M}}{\mathcal{N}}\right)\right], \quad (7)$$

where $f(.)$ is a convex function and as per the definition, $f(1) = 0$. Using (7) and Jensen's inequality, the following relation can be deducted:

$$E_{\mathcal{N}}\left[f\left(\frac{\mathcal{M}}{\mathcal{N}}\right)\right] \geq f\left[E_{\mathcal{N}}\left(\frac{\mathcal{M}}{\mathcal{N}}\right)\right]$$

$$\Rightarrow D_f(\mathcal{M} \,||\, \mathcal{N}) \geq 0 \qquad \left[\text{as } f\left(\int_x \mathcal{M}(x)dx\right) = f(1) = 0\right]. \quad (8)$$

The authors are with the Biomedical Imaging and Bioinformatics Lab, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. E-mail: {sumancse_r, pmaji}@isical.ac.in.

The minimum value of $D_f(\mathcal{M} \;||\; \mathcal{N})$ is 0 and is achieved if and only if $\mathcal{M} = \mathcal{N}$. So, $\chi^2_{Pearson}\left[2P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s} \;||\; (P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}} + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s})\right]$ achieves 0 if and only if:

$$2P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s) = P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) + P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s)$$
$$\Rightarrow P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}(x, z_c, z_s) = P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}(x, z_c, z_s). \quad (9)$$

So, theoretically, joint distribution corresponding to generated/fake encoding becomes identical to the joint distribution corresponding to real encoding. As a result, the generator $\mathcal{G}$ becomes successful to beat the discriminator $\mathcal{D}$, that is, the discriminator cannot determine whether the triplet (x, z_c, z_s) is coming from real joint distribution $P_{\mathcal{E}_c\mathcal{E}_s\mathcal{X}}$ or generated/fake joint distribution $P_{\mathcal{G}\mathcal{Z}_c\mathcal{Z}_s}$, and hence, it concludes the proof.

SII. Computation of Regularization Terms

Based on the assumption that the prior $P_{\mathcal{Z}_s}(z_s)$ for latent stain density code z_s is a standard normal distribution, the KL divergence between $Q(z_s)$ and $P_{\mathcal{Z}_s}(z_s)$ in sub-expression R_2 of (16) of the main manuscript can be computed as follows:

$$D_{KL}[Q(z_s) \;||\; P_{\mathcal{Z}_s}(z_s)] = \frac{1}{2}\log\left(\frac{|\Sigma_P|}{|\Sigma_Q|}\right) - \frac{k}{2} +$$
$$\frac{1}{2}\,\mathrm{Tr}\left(\Sigma_P^{-1}\Sigma_Q\right) + \frac{1}{2}(\mu_Q - \mu_P)^T\Sigma_P^{-1}(\mu_Q - \mu_P), \quad (10)$$

where Σ_P and Σ_Q represent covariance matrices corresponding to prior distribution $P_{\mathcal{Z}_s}(z_s)$ and auxiliary distribution $Q(z_s)$, respectively, k denotes the dimensionality of the stain density code, μ_Q and μ_P denote the mean values corresponding to $Q(z_s)$ and $P_{\mathcal{Z}_s}(z_s)$, respectively.

Now, to compute the KL divergence between color appearance prior $P_{\mathcal{Z}_c}(z_c)$ and auxiliary distribution $Q(z_c)$, the KL divergence between two mixture models is to be computed. But, unfortunately, there is no closed form solution for computing KL divergence between two truncated normal mixture models. Based on the variational approximation method proposed in [1], the KL divergence in sub-expression R_1 of (16) of the main manuscript is approximated, which requires computation of KL divergence between individual mixture components: individual truncated normal distributions. The KL divergence between two d-dimensional truncated normal distributions $\mathbb{F}$ and $\mathbb{G}$ is computed as follows:

$$D_{KL}(\mathbb{F} \;||\; \mathbb{G}) = E_{\mathbb{F}}\left[\log\left(\frac{\mathbb{F}(x; \mu_f, \Sigma_f, a_f, b_f)}{\mathbb{G}(x; \mu_g, \Sigma_g, a_g, b_g)}\right)\right]$$
$$\Rightarrow D_{KL}(\mathbb{F} \;||\; \mathbb{G}) = \log\left(\frac{\mathcal{A}_{\mathbb{G}}}{\mathcal{A}_{\mathbb{F}}}\right) - \frac{d}{2} + \frac{1}{2}\,\mathrm{Tr}\left[\Sigma_g^{-1}\Sigma_f\right]$$
$$+\frac{1}{2}(\mu_f - \mu_g)^T\Sigma_g^{-1}(\mu_f - \mu_g), \quad (11)$$

$$\text{where } \mathcal{A}_{\mathbb{F}} = \int_{a_f}^{b_f} \exp\left[-\frac{1}{2}\{(z - \mu_f)^T\Sigma_f^{-1}(z - \mu_f)\}\right] dz \quad (12)$$

$$\text{and } \mathcal{A}_{\mathbb{G}} = \int_{a_g}^{b_g} \exp\left[-\frac{1}{2}\{(z - \mu_g)^T\Sigma_g^{-1}(z - \mu_g)\}\right] dz. \quad (13)$$

where Σ_f and Σ_g represent covariance matrices corresponding to distributions $\mathbb{F}$ and $\mathbb{G}$, respectively, μ_f and μ_g denote the mean values corresponding to $\mathbb{F}$ and $\mathbb{G}$, respectively.

SIII. Basics of Generative Adversarial Networks

This section provides a brief description of the generative framework of GAN, which forms the basis of the proposed deep generative model. Being inspired by game theoretic approach zero-sum game, where one's loss is other's win, Goodfellow *et al.* proposed GAN model in [2]. The basic GAN framework is comprised of two sub-networks: a generator ($\mathcal{G}$) and a discriminator ($\mathcal{D}$), which compete adversarially against each other. The discriminator network is employed to discriminate between real data, sampled from real data distribution $P_{data}(x)$ and generated data. The task of the generator is to learn a data distribution, which is identical to the real data distribution so that given a noisy input z, sampled from a random distribution $P_{\mathcal{Z}}(z)$, it can generate data as close as possible to the real data samples. The objective function corresponding to the zero-sum game is framed as a min-max optimization problem as follows:

$$\max_{\mathcal{D}}\min_{\mathcal{G}} V(\mathcal{D}, \mathcal{G}) = \max_{\mathcal{D}}\min_{\mathcal{G}} E_{x\sim P_{data}(x)}[\log \mathcal{D}(x)]$$
$$+E_{z\sim P_{\mathcal{Z}}(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)))], \quad (14)$$

where E is the expectation operator.

With a proper training, the above formulation enforces the generator network to learn a probability distribution identical to the real data distribution without applying any parametric estimation method. As a result, GAN has earned acceptance in diverse domains and has been applied in different types of problems. But, the loss function of GAN suffers from the following problems: (i) the vanishing gradient problem: initially when the generated data sample is easily beatable by the discriminator then the term $\log(1 - \mathcal{D}(\mathcal{G}(z)))$ tends to 0 and as a result, the derivative produces 0 value, which does not help in the updation of the generator parameters; and (ii) unstable training problem: in one step, the objective function is maximized with respect to discriminator $\mathcal{D}$ and in the next step, the same objective function is minimized with respect to generator $\mathcal{G}$, and thus, the training process becomes unstable. To deal with the above problems and to ensure stable convergence, Mao *et al.* introduced a least square loss function based GAN (LSGAN) in [3].

SIV. Algorithms: Training & Mapping

In this section, the relevant algorithms are provided. Algorithm 1 summarizes the training procedure for the proposed method. Once the networks are trained, Algorithm 2 is used to generate color normalized source images based on the trained network parameters $\{\theta_{\mathcal{E}_c}, \theta_{\mathcal{E}_s}, \theta_{\mathcal{G}}\}$, obtained via Algorithm 1.

SV. Computation of Different Evaluation Indices

To evaluate the performance of different stain estimation and separation methods, standard deviation, symmetric Kullback-Leibler (KL) divergence and signal-to-noise ratio (SNR) are used. The details of the computation of different evaluation indices are described in details in the following subsections.

Algorithm 1 Mini-batch stochastic gradient descent training of proposed deep generative model.

Input: Training set images sampled from real data distribution $P_X(x)$, number of training epochs (τ).
Output: Trained parameter set $\{\theta_{\mathcal{D}}, \theta_{\mathcal{E}_c}, \theta_{\mathcal{E}_s}, \theta_{\mathcal{G}}\}$, corresponding to four deep networks $\mathcal{D}$, $\mathcal{E}_c$, $\mathcal{E}_s$ and $\mathcal{G}$, respectively.

1: **for** each epoch $t = 0$ to $\tau - 1$ **do**

- Sample mini-batch of M images $\{x_m\}_{m=1}^{M}$ from real data distribution $P_X(x)$.
- Generate M color appearance codes $\{z_{c_m}\}_{m=1}^{M}$ by using $\mathcal{E}_c(x_m; \theta_{\mathcal{E}_c})$.
- Generate M stain density codes $\{z_{s_m}\}_{m=1}^{M}$ by using $\mathcal{E}_s(x_m; \theta_{\mathcal{E}_s})$.
- Generate reconstructed images $\{\hat{x}_m\}_{m=1}^{M}$ by using $\mathcal{G}(z_{c_m}, z_{s_m}; \theta_{\mathcal{G}})$.
- Sample mini-batch of M color appearance codes $\{\acute{z}_{c_m}\}_{m=1}^{M}$ from latent color appearance prior $P_{\mathcal{Z}_c}(z_c)$.
- Sample mini-batch of M stain density codes $\{\acute{z}_{s_m}\}_{m=1}^{M}$ from latent stain density prior $P_{\mathcal{Z}_s}(z_s)$.
- Update discriminator $\mathcal{D}$ by descending in the direction of stochastic gradient:

$$\nabla_{\theta_{\mathcal{D}}} \left(\frac{1}{M} \sum_{m=1}^{M} \left[(A - \mathcal{D}(x_m, z_{c_m}, z_{s_m}; \theta_{\mathcal{D}}))^2 + (B - \mathcal{D}(\hat{x}_m, \acute{z}_{c_m}, \acute{z}_{s_m}; \theta_{\mathcal{D}}))^2 \right] \right)$$

- Update color appearance encoder $\mathcal{E}_c$ by descending in the direction of stochastic gradient:

$$\nabla_{\theta_{\mathcal{E}_c}} \left(\frac{1}{M} \sum_{m=1}^{M} \left[(A - \mathcal{D}(x_m, z_{c_m}, z_{s_m}; \theta_{\mathcal{D}}))^2 + D_{KL}(z_{c_m} \,||\, \acute{z}_{c_m}) \right] \right)$$

- Update stain density encoder $\mathcal{E}_s$ by descending in the direction of stochastic gradient:

$$\nabla_{\theta_{\mathcal{E}_s}} \left(\frac{1}{M} \sum_{m=1}^{M} \left[(A - \mathcal{D}(x_m, z_{c_m}, z_{s_m}; \theta_{\mathcal{D}}))^2 + D_{KL}(z_{s_m} \,||\, \acute{z}_{s_m}) \right] \right)$$

- Update decoder/generator $\mathcal{G}$ by descending in the direction of stochastic gradient:

$$\nabla_{\theta_{\mathcal{G}}} \left(\frac{1}{M} \sum_{m=1}^{M} \left[(C - \mathcal{D}(\hat{x}_m, \acute{z}_{c_m}, \acute{z}_{s_m}; \theta_{\mathcal{D}}))^2 - \log(\hat{x}_m) \right] \right)$$

2: Stop.

Algorithm 2 Color normalization of source images via color appearance mapping.

Input: A set of N source images $\{x_n^S\}_{n=1}^{N}$, template image x^T, trained network parameters $\{\theta_{\mathcal{E}_c}, \theta_{\mathcal{E}_s}, \theta_{\mathcal{G}}\}$ corresponding to deep networks $\mathcal{E}_c$, $\mathcal{E}_s$ and $\mathcal{G}$, obtained via Algorithm 1.
Output: Color normalized source images $\{\tilde{x}_n^S\}_{n=1}^{N}$.

1: Generate latent color appearance code z_c^T and latent stain density code z_s^T corresponding to template image x^T by using $\mathcal{E}_c(x^T; \theta_{\mathcal{E}_c})$ and $\mathcal{E}_s(x^T; \theta_{\mathcal{E}_s})$, respectively.

2: **for** each image in source image set $\{x_n^S\}_{n=1}^{N}$ **do**

- Generate latent color appearance code $z_{c_n}^S$ and latent density code $z_{s_n}^S$ by using $\mathcal{E}_c(x_n^S; \theta_{\mathcal{E}_c})$ and $\mathcal{E}_s(x_n^S; \theta_{\mathcal{E}_s})$, respectively.
- Feed template image latent color appearance code z_c^T and source image latent stain density code $z_{s_n}^S$ to the generator $\mathcal{G}$ and by using $\mathcal{G}(z_c^T, z_{s_n}^S; \theta_{\mathcal{G}})$ generate normalized source image $\tilde{x}_n^S$

3: Stop.

A. Standard Deviation

It is reported in [4] that UCSB data set consists of 58 H&E stained histological images, grouped among 10 biopsy sets. The information regarding 10 biopsy sets is reported in Table S1. It is expected that the images within same biopsy set have been subjected through similar staining routine and identical storage condition. Using different stain separation methods, corresponding stain color appearance matrices of dimension $3 \times c$ are estimated, where c represents the number of stains involved in the staining routine. Each column of the stain color appearance matrix represents a separate stain vector. A good spectral estimation algorithm should produce consistent stain vectors for the images within the same biopsy set.

The evaluation of spectral estimation by element-wise standard deviation (σ) measure was originally reported in [5]. As the images within the same biopsy set are expected to produce consistent stain vectors, smallest element-wise standard deviation value denotes best spectral estimation algorithm. The general representation of the stain color appearance matrix for Hematoxylin (H) and Eosin (E) in H&E stained images is given as follows:

$$\begin{bmatrix} r_{\mathrm{H}} & r_{\mathrm{E}} \\ g_{\mathrm{H}} & g_{\mathrm{E}} \\ b_{\mathrm{H}} & b_{\mathrm{E}} \end{bmatrix}$$

where the first column and second column represent the stain color appearance vectors in RGB domain corresponding to H-stain and E-stain, respectively.

The element-wise standard deviations of the estimated H-stain vectors within the same biopsy set are computed, which are represented as σ_{H}. Similarly, for E stain, the element-wise

TABLE S1
TEN BIOPSY SETS AND RELATED IMAGES OF UCSB DATA SET

Biopsy Set	Images	Biopsy Set	Images
ytma10_010704	ytma10_010704_benign1 ytma10_010704_benign2 ytma10_010704_benign3 ytma10_010704_malignant1 ytma10_010704_malignant2 ytma10_010704_malignant3	ytma49_042403	ytma49_042403_benign1 ytma49_042403_benign2 ytma49_042403_benign3 ytma49_042403_malignant1 ytma49_042403_malignant2 ytma49_042403_malignant3
ytma12_010804	ytma12_010804_benign1 ytma12_010804_benign2 ytma12_010804_benign3 ytma12_010804_malignant1 ytma12_010804_malignant2 ytma12_010804_malignant3	ytma49_072303	ytma49_072303_benign1 ytma49_072303_benign2 ytma49_072303_malignant1 ytma49_072303_malignant2
ytma23_022103	ytma23_022103_benign1 ytma23_022103_benign2 ytma23_022103_benign3 ytma23_022103_malignant1 ytma23_022103_malignant2 ytma23_022103_malignant3	ytma49_111003	ytma49_111003_benign1 ytma49_111003_benign2 ytma49_111003_benign3 ytma49_111003_malignant1 ytma49_111003_malignant2 ytma49_111003_malignant3
ytma49_042003	ytma49_042003_benign1 ytma49_042003_benign2 ytma49_042003_benign3 ytma49_042003_malignant1 ytma49_042003_malignant2 ytma49_042003_malignant3	ytma49_111303	ytma49_111303_benign1 ytma49_111303_benign2 ytma49_111303_benign3 ytma49_111303_malignant1 ytma49_111303_malignant2 ytma49_111303_malignant3
ytma49_042203	ytma49_042203_benign1 ytma49_042203_benign2 ytma49_042203_benign3 ytma49_042203_malignant1 ytma49_042203_malignant2 ytma49_042203_malignant3	ytma55_030603	ytma55_030603_benign1 ytma55_030603_benign2 ytma55_030603_benign3 ytma55_030603_benign4 ytma55_030603_benign5 ytma55_030603_benign6

standard deviation vector is computed and represented as σ_{E}. The procedure can be illustrated using an example. It can be observed from Table S1 that biopsy set ytma10_010704 contains 6 H&E stained images. The stain color appearance matrix corresponding to each of the 6 images is estimated. The column vectors designating the H-stain color appearance are extracted as H-stain vectors and element-wise standard deviation corresponding to 6 vectors are computed and reported as σ_{H} corresponding to the biopsy set ytma10_010704. Similar computation is done for E-stain also and the element-wise standard deviation vector is reported as σ_{E}.

B. Symmetric Kullback-Leibler (KL) Divergence

The CMU data set [6] consists of only 3 images, but it provides the ground truth stain separated images (H-stain and E-stain), corresponding to each H&E stained histological images. Based on different stain separation methods, corresponding stain color appearance matrices are estimated. The image corresponding to i-th stain is generated using the Beer-Lambert law of colorimetry [7]:

$$I_i = I^b \exp(-M^*(:,i) \times D^*(i,:)) \tag{15}$$

Here, I_i denotes decomposition image corresponding to i-th stain, I^b represents the intensity value of a background pixel, $M^*(:,i)$ represents the i-th column vector of color appearance matrix M^* and represents the i-th stain vector. $D^*(i,:)$ denotes the i-th row of the stain density map D^*.

TABLE S2
PERFORMANCE OF DIFFERENT INDIVIDUAL MODELS, LATENT REPRESENTATION BASED VARIANTS, AND COLOR NORMALIZATION METHODS ON UCSB DATA SET USING NMI, BICC, AND WSCC: THE HIGHEST VALUES ARE MARKED IN BOLD FONT.

Different Methods	NMI		BiCC		WsCC	
	Mean	Median	Mean	Median	Mean	Median
B+C+D+E	0.7221	0.7273	0.6920	0.6906	0.5036	0.4885
A+C+D+E	0.7234	0.7079	0.6919	0.6818	0.5115	0.4921
A+B+D+E	0.7252	0.7266	0.6913	0.6847	0.5059	0.5013
A+B+C+E	0.7404	0.7524	0.7117	0.7128	0.5304	0.5248
A+B+C+D	0.7363	0.7378	0.7044	0.7027	0.5232	0.5276
z_c&z_s correlated	0.7273	0.7275	0.6989	0.7018	0.5115	0.5162
C-L+D-R	0.7433	0.7515	0.7114	0.7184	0.5318	0.5326
C-L+D-L	0.7143	0.7138	0.6793	0.6786	0.4877	0.4924
C-R+D-R	0.7371	0.7363	0.7054	0.6999	0.5199	0.5108
GMM+NSGAN	0.7326	0.7234	0.7010	0.6955	0.5180	0.5020
GMM+LSGAN	0.7510	0.7540	0.7227	0.7206	0.5475	0.5319
tGMM+NSGAN	0.7497	0.7487	0.7199	0.7104	0.5442	0.5381
Unimodal	0.4078	0.3926	0.3604	0.3556	0.1506	0.1394
ColTrans	0.6620	0.6810	0.6134	0.6256	0.4109	0.4262
PF	0.6895	0.6890	0.6578	0.6609	0.4552	0.4540
EPF	0.6816	0.6829	0.6484	0.6506	0.4438	0.4474
SCD	0.6029	0.5861	0.5664	0.5637	0.3471	0.3351
HTN	0.7069	0.7016	0.6745	0.6611	0.4776	0.4579
SPCN	0.6802	0.6794	0.6476	0.6461	0.4416	0.4397
SN-GAN	0.7069	0.6971	0.6702	0.6673	0.4743	0.4708
StainGAN	0.6632	0.6579	0.6255	0.6215	0.4174	0.4154
AST	0.6846	0.6816	0.6439	0.6440	0.4418	0.4486
RFCC	0.7126	0.7071	0.6925	0.6813	0.4925	0.4731
TredMiL	**0.7547**	**0.7620**	**0.7281**	**0.7332**	**0.5600**	**0.5557**

As both the ground truth and estimated images corresponding to each stain are present, Kullback-Leibler (KL) divergence measure is used to evaluate the performance of different stain separation methods. The standard KL divergence measure deals with two probability distributions. So, the normalized hue histograms corresponding to ground truth and estimated stain image are represented as two probability distributions in this case, and are subjected to the KL divergence measure, which is defined as follows:

$$D_{KL}(P||Q) = \sum_{h \in H} P(h) \log \left(\frac{P(h)}{Q(h)} \right) \tag{16}$$

where h represents one hue value, which denotes a single bin in the whole hue histogram, represented by H. P and Q represent the two discrete probability distributions. As KL divergence is a $\log$ based measure, a low KL divergence value denotes better reconstruction. Now, as KL divergence is not a symmetric measure, here KL divergence is computed both-ways: assuming ground truth image normalized hue histogram (say, H_{GT}) and estimated image normalized hue histogram (say, H_{EST}) as P and Q, respectively and vice versa and the average of these two KL divergence measures are reported as symmetric KL divergence measure (say, KL_{sym}):

$$KL_{\mathrm{sym}} = \frac{D_{KL}(H_{\mathrm{GT}}||H_{\mathrm{EST}}) + D_{KL}(H_{\mathrm{EST}}||H_{\mathrm{GT}})}{2}. \tag{17}$$

C. Signal-to-Noise Ratio (SNR)

As discussed in Section Sv-B, the images corresponding to H-stain and E-stain are estimated from the H&E stained histological images. The R, G, B channels corresponding

to ground truth stain image and estimated stain image are extracted and they are designated as GT_r, GT_g, GT_b and EST_r, EST_g, EST_b, respectively. The signal power (say, SP_k) and noise power (say, NP_k) corresponding to k-th image channel ($k \in \{\text{r, g, b}\}$) are computed as follows:

$$SP_k = \sum_i \sum_j (\text{GT}_k(i,j))^2;$$

$$NP_k = \sum_i \sum_j (\text{GT}_k(i,j) - \text{EST}_k(i,j))^2. \quad (18)$$

The SNR corresponding to k-th channel ($k \in \{\text{r, g, b}\}$) is computed as follows:

$$SNR_k = 10 \times \log_{10}\left(\frac{SP_k}{NP_k}\right). \quad (19)$$

The final SNR value is computed by averaging the SNR values corresponding to R, G, B channels as follows:

$$SNR = \frac{SNR_{\text{r}} + SNR_{\text{g}} + SNR_{\text{b}}}{3}. \quad (20)$$

As good reconstruction always makes the denominator NP_k small, a high SNR value always designates a better algorithm.

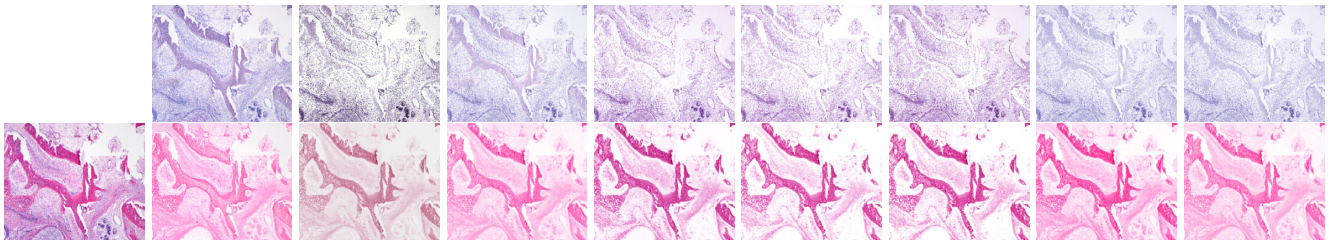
SVI. Experimental Setup for Existing Algorithms

The performance of the proposed rough-fuzzy circular clustering based stain separation and color normalization method is studied extensively and compared with that of

- several state-of-the-art stain separation methods, namely, plane fitting (PF) [8], enhanced plane fitting (EPF) [6], HTN method [5] structure-preserving color normalization (SPCN) [9], expectation-maximization (EM) based circular clustering algorithm [10], RFCC method [11]; and
- different color normalization algorithms such as color transfer technique (ColTrans) [12], stain color description (SCD) [13], stain normalization using generative adversarial networks (SN-GAN) [14], StainGAN [15], and adversarial stain transfer (AST) [16], as well as PF [8], EPF [6], HTN [5], SPCN [9] and RFCC [11].

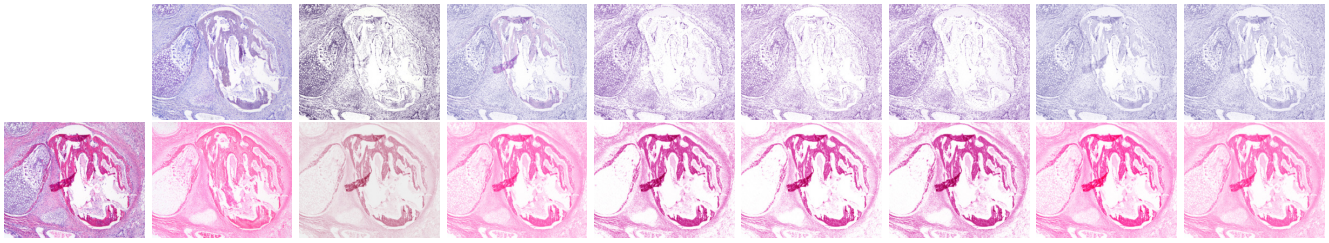
The experimental setup used for each of the existing algorithms is briefly outlined next:

- **ColTrans [12]:** The source code of ColTrans is downloaded from https://warwick.ac.uk/fac/sci/dcs/research/tia/software/sntoolbox and it is simulated using MATLAB R2014a (8.3.0.532).
- **PF [8]:** There are three parameters involved in the PF approach. The OD threshold for transparent pixels (β) is taken to be 0.15, the tolerance value for pseudo-min and pseudo-max (α) is considered to be 1 and the transmitted light intensity (I_0) is considered to be 255. For each channel of the image, 99^{th} percentile of the stain concentration is computed. Stain matrix in this method must be of dimension [2 × 3] or [3 × 3]. If only two rows are present, then the third row is estimated as the cross product of the first two rows. The code is downloaded from https://warwick.ac.uk/fac/sci/dcs/research/tia/software/sntoolbox and it is simulated using MATLAB R2014a (8.3.0.532).
- **EPF [6]:** The source code is downloaded from http://jelena.ece.cmu.edu/repository/rr/14_McCannMPCK/14_McCannMPCK.html and it is simulated using MATLAB R2014a (8.3.0.532). Most of the parameters are identical with that of the PF method. OD threshold for transparent pixels (β) is taken to be 0.15, the tolerance value for pseudo-min and pseudo-max (α) is considered to be 1 and the transmitted light intensity (I_0) here is considered to be 240. For each channel of the image, 99^{th} percentile of the stain concentration is computed. Stain matrix in this method must be of dimension $[2 \times 3]$ or $[3 \times 3]$. If only two rows are present, then the third row is estimated as the cross product of the first two rows. The threshold for removing totally white pixels is considered to be $1E-3$.
- **SCD [13]:** In this method, transmitted light intensity (I_0) is considered to be 255. The background probability threshold (T_{bgd}) and stain probability threshold (T_{fgd}) are considered to be 0.75 and 0.75 respectively. The background label is considered as 0 for the ease of implementation. The code is downloaded from https://warwick.ac.uk/fac/sci/dcs/research/tia/software/sntoolbox and it is simulated using MATLAB R2014a (8.3.0.532).
- **HTN [5]:** The number of stains involved in an image is taken to be 2, hue value corresponding to each of the stains is within the range $[0, 360)$. The initial hue center values for Hematoxylin (H) and Eosin (E) are taken to be 260 and 320 respectively. The filter size for illuminant normalization module is taken to be 5 and the tolerance value for non-negative matrix factorization is considered to be $1E-06$. Reference illuminant matrix is taken to be $[255, 255, 255]$. The corresponding code is downloaded from https://www.comm.utoronto.ca/∼xyli/index.php?page=Publication and the code is simulated in MATLAB R2014a (8.3.0.532).
- **SPCN [9]:** The number of stains is considered to be 2, weightage to sparsity regularization constraint (λ) is taken to be 0.1. Here, λ value of 0 designates standard NMF. The source code is downloaded from https://github.com/abhishekvahadane/CodeRelease_ColorNormalization and the code is simulated in MATLAB R2014a (8.3.0.532). As per the code, the maximum number of iterations allowed is restricted to be 200.
- **EM [10]:** The number of stains in an image is taken to be 2 (ideally) or 3, hue value corresponding to each of the stains is within the range $[0, 360)$. The initial hue center values for H and E stains are considered to be 260 and 320 respectively. The filter size for illuminant normalization module is taken to be 5. The tolerance value for non-negative matrix factorization is considered to be $1E-06$ and reference illuminant matrix is taken to be $[255, 255, 255]$. The corresponding code is also downloaded from https://www.comm.utoronto.ca/∼xyli/index.php?page=Publication and the code is executed in MATLAB R2014a (8.3.0.532). As per the code, the maximum number of iterations allowed is 500. The initial



(a) CMU image (b) GT (c) PF (d) EPF (e) HTN (f) SPCN (g) EM (h) RFCC (i) TredMiL

Fig. S1. CMU image_2, ground-truth stain spectra, and estimated stain spectra using different stain separation methods (top: H-stain; bottom: E-stain)



(a) CMU image (b) GT (c) PF (d) EPF (e) HTN (f) SPCN (g) EM (h) RFCC (i) TredMiL

Fig. S2. CMU image_3, ground-truth stain spectra, and estimated stain spectra using different stain separation methods (top: H-stain; bottom: E-stain)

cluster centroid (μ) and concentration parameter (κ) are estimated as per the initialization method stated in [10].

- **SN-GAN [14]:** For the source code, we have used the InfoGAN [17] implementation and the code is downloaded from https://github.com/openai/InfoGAN and tuned the parameters as suggested in [14]. The code is executed using Python3. The lightness channel in generator network maps to a latent k-simplex probability subspace. Here, k is considered to be 3 for hematoxylin, eosin and background regions in H&E images by using a softmax layer. For measuring the reconstruction quality, L2 loss is utilized. The momentum factor for Adam optimizer (β) is considered to be 0.5 with fixed learning rate (lr) 0.0001. The model is trained on 299×299 randomly cropped patches and evaluated on full size images by using leave-one-out cross-validation.
- **StainGAN [15]:** The source code is downloaded from https://xtarx.github.io/StainGAN/ and the code is executed using Python3 and pyTorch. The maximum number of iterations allowed is 100, the momentum term for Adam optimizer (β) is considered to be 0.5, the batch size is taken to be 4. The size of the image buffer that stores previously generated images is considered to be 50. The number of generating filters and discriminating filters in first convolutional layer are taken to be 64 and 64, respectively and the initial learning rate (lr) is taken to be 0.0002. Here, as CPU based system is used for execution, the default GPU ID is taken to be -1.
- **AST [16]:** The source code is downloaded from http://www.sfu.ca/∼abentaie/stain_tf/stain_tf.html. It is a Torch based implementation, and for executing the source code, Python3 and Lua (version 5.2) is used. As per the implementation, the batch size is taken to be 100, the maximum number of iterations allowed is 100, the learning rate (lr) is 0.00002 and the momentum factor for Adam optimizer (β) is considered to be 0.5. As the code is executed in a CPU based system, the GPU ID is considered to be 0.
- **RFCC [11]:** In this method, for H&E stained histological image sets, the number of stain classes are considered to be 2 (ideally) or 3 (considering the achromatic region). For each image, the initial cluster centroids, corresponding to the stain classes, are computed using circular thresholding method, proposed in [18]. For clustering analysis, the fuzzifier value (m) is chosen to 2.0, the tolerance value is set to be $1E-4$. The threshold value λ is computed adaptively based on the highest and second highest membership value of each pixel to stain classes. Total number of iterations are chosen to be 100. During the computation of local neighborhood information, the size of neighborhood window is chosen to be 3×3.

SVII. SOME RESULTS AND DISCUSSION

Table S2 reports the mean and median values with respect to the color constancy indices, NMI, BiCC and WsCC, respectively, corresponding to different color normalization approaches. The corresponding p-values, with respect to Wilcoxon signed rank test and paired-*t* test, are reported in Table I of the main article. From the results reported in Table S2, it can be observed that the proposed model TredMiL attains highest NMI, BiCC and WsCC values (marked in bold font) among all the color normalization approaches.

Fig. S1 and S2 depict the qualitative comparison of stain separation by different methods. From Fig. S1 and S2, it is evident that only the proposed method, RFCC and EPF can extract the intrinsic structures of the biological components, highlighted by H-stain and E-stain. But, the main disadvantage of the EPF method is that it is not applicable in the situation where more than two stains exist. In RFCC method, clustering analysis is performed on an approximated hue histogram, which cannot capture spatial relationship among the neighborhood pixels in an image. Also, to ensure non-negativity of

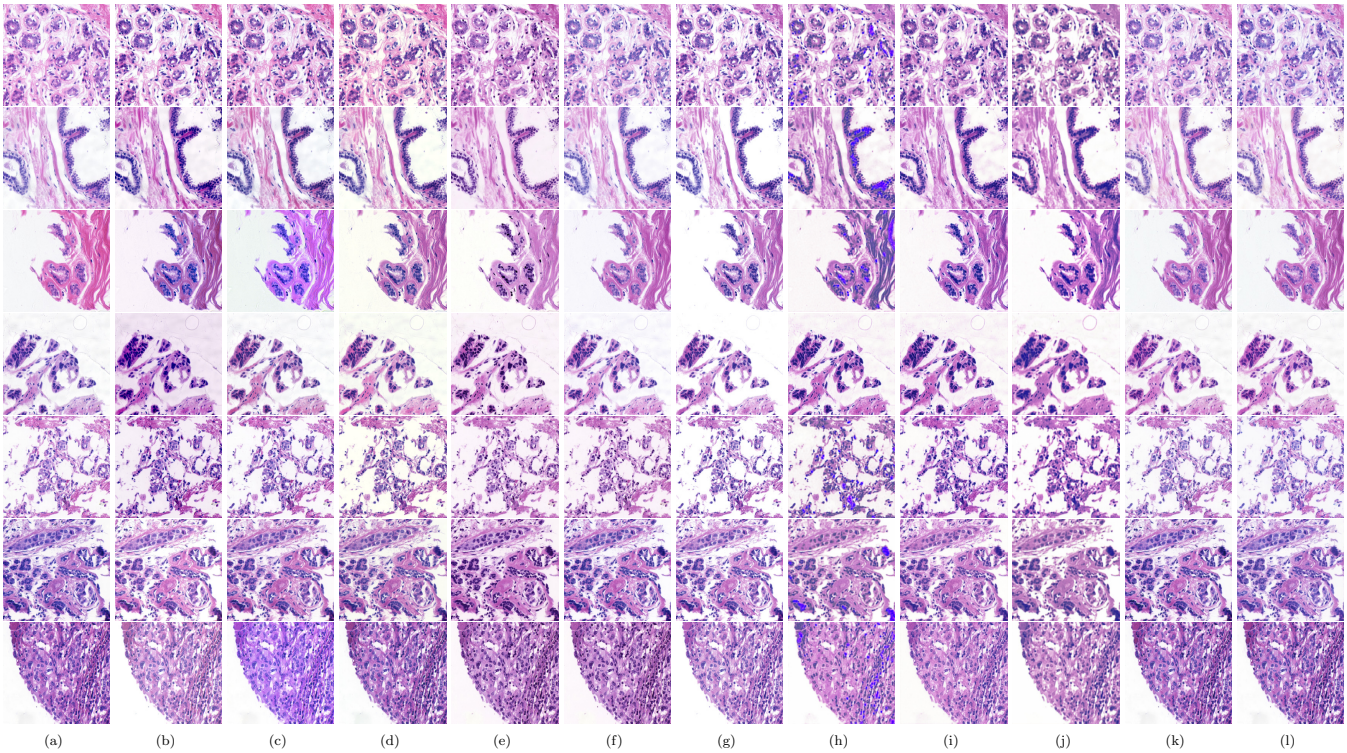


Fig. S3. (a) Original images of UCSB data set; and color normalized images obtained using different color normalization methods: (b) ColTrans, (c) PF, (d) EPF, (e) SCD, (f) HTN, (g) SPCN, (h) SN-GAN, (i) StainGAN, (j) AST, (k) RFCC and (l) TredMiL

the stain color appearance matrix and associated stain density map, RFCC uses NMF method, which suffers from unstable convergence problem. The qualitative performance analysis of different color normalization methods on UCSB data set is presented in Fig. S3. From the results reported in Fig. S3, it can be observed that the proposed TredMiL model performs better than other state-of-the-art color normalization methods as per the color consistency after normalization is concerned.

REFERENCES

[1] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, 2007, pp. 317–320.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 2672–2680.

[3] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "On the Effectiveness of Least Squares Generative Adversarial Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 2947–2960, 2019.

[4] E. D. Gelasca, J. Byun, B. Obara, and B. S. Manjunath, "Evaluation and Benchmark for Biological Image Segmentation," in *Proceedings of IEEE International Conference on Image Processing*, 2008, pp. 1816–1819.

[5] X. Li and K. N. Plataniotis, "A Complete Color Normalization Approach to Histopathology Images Using Color Cues Computed from Saturation-Weighted Statistics," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 7, pp. 1862–1873, 2015.

[6] M. T. McCann, J. Majumdar, C. Peng, C. A. Castro, and J. Kovačević, "Algorithm and Benchmark Dataset for Stain Separation in Histology Images," in *Proceedings of IEEE International Conference on Image Processing*, 2014, pp. 3953–3957.

[7] W. W. Parson, *Modern Optical Spectroscopy*. Springer, 2007.

[8] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, "A Method for Normalizing Histology Slides for Quantitative Analysis," in *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009, pp. 1107–1110.

[9] A. Vahadane, T. Peng, A. Sethi, S. Albarqouni, L. Wang, M. Baust, K. Steiger, A. M. Schlitter, I. Esposito, and N. Navab, "Structure-Preserving Color Normalization and Sparse Stain Separation for Histological Images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 8, pp. 1962–1971, 2016.

[10] X. Li and K. Plataniotis, "Circular Mixture Modeling of Color Distribution for Blind Stain Separation in Pathology Images," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 150–161, 2017.

[11] P. Maji and S. Mahapatra, "Circular Clustering in Fuzzy Approximation Spaces for Color Normalization of Histological Images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1735–1745, 2020.

[12] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color Transfer Between Images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.

[13] A. M. Khan, N. Rajpoot, D. Treanor, and D. Magee, "A Nonlinear Mapping Approach to Stain Normalization in Digital Histopathology Images Using Image-Specific Color Deconvolution," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1729–1738, 2014.

[14] F. G. Zanjani, S. Zinger, B. E. Bejnordi, J. A. W. M. van der Laak, and P. H. N. de With, "Stain Normalization of Histopathology Images Using Generative Adversarial Networks," in *Proceedings of IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 573–577.

[15] M. T. Shaban, C. Baur, N. Navab, and S. Albarqouni, "StainGAN: Stain Style Transfer for Digital Histological Images," in *Proceedings of IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 953–956.

[16] A. Bentaieb and G. Hamarneh, "Adversarial Stain Transfer for Histopathology Image Analysis," *IEEE Transactions on Medical Imaging*, vol. 37, no. 3, pp. 792–802, 2018.

[17] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 2180–2188.

[18] Y. K. Lai and P. L. Rosin, "Efficient Circular Thresholding," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 992–1001, 2014.