



Self-supervision Based Dual-Transformation Learning for Stain Normalization, Classification and Segmentation

Shiv Gehlot^(✉) and Anubha Gupta

SBILab, Department of ECE, Indraprastha Institute of Information Technology-Delhi
(IIIT-D), Delhi, India
{shivg,anubha}@iiitd.ac.in

Abstract. Stain color variations across images are common in the medical imaging domain. However, such variations among the training and test datasets may lead to unsatisfactory performance on the latter in any desired task. This paper proposes a novel coupled-network composed of two U-Net type architectures that utilize self-supervised learning. The first subnetwork (N1) learns an identity transformation, while the second (N2) learns a transformation to perform stain normalization. We also introduce classification heads in the subnetworks, trained along with the stain normalization task. To the best of our knowledge, the proposed coupling framework, where the information from the encoders of both the subnetworks is utilized by the decoders of both subnetworks as well as trained in a coupled fashion, is introduced in this domain for the first time. Interestingly, the coupling of N1 (for identity transformation) and N2 (for stain normalization) helps N2 learn the stain normalization task while being cognizant of the features essential to reconstruct images. Similarly, N1 learns to extract relevant features for reconstruction invariant to stain color variations due to its coupling with N2. Thus, the two subnetworks help each other, leading to improved performance on the subsequent task of classification. Further, it is shown that the proposed architecture can also be used for segmentation, making it applicable for all three applications: stain normalization, classification, and segmentation. Experiments are carried out on four datasets to show the efficacy of the proposed architecture.

Keywords: Self-supervised learning · Stain-normalization · Classification

1 Introduction

Staining is a step in microscopic slide preparation that involves staining chemicals to highlight the regions of interest in the tissue/blood sample or bone mar-

Shiv Gehlot would like to thank University Grant Commission (UGC), Govt. of India for the UGC-Senior Research Fellowship. We also acknowledge the Infosys Center for Artificial Intelligence, IIIT-Delhi for our research work.

row smear. Stain color variations in the images across datasets collected from various hospitals/centers may arise due to varying illumination conditions, stain chemicals, and staining time. Thus, the diagnostic tool trained on one center's dataset may perform non-optimally on another center's dataset. To counter this problem, stain color normalization is employed. Some of the widely used stain normalization approaches are based on histogram equalization [6, 8, 18], color deconvolution [9, 11, 15, 17], and color transfer methods [10, 13, 14]. In histogram equalization methods, the probability density function of the reference image and source image colors are matched for each R, G, and B channel. In [10], color transfer between the two images is used to achieve stain normalization, while in [13], mean and standard deviation of the source and reference images are matched. However, these methods do not utilize histological information. Color deconvolution methods are based on negative matrix factorization (NMF) or singular value decomposition (SVD) to find the stain vectors. In [4] SVD, along with the robust aligning of the reference and query images' Cartesian frames is utilized to counter stain color variation. A limitation of all these methods is that the performance depends on the reference image's choice and may change considerably with the change of reference image. Of late, deep learning (DL) methods are gaining significance in this area. Some generative model-based DL methods for stain normalization are discussed in [16, 21, 22]. In [22], InfoGAN is utilized, while in [21], variational-autoencoder and deep convolutional Gaussian mixture model are used for stain normalization. In [16], a CycleGAN based method is used to preserve the structure information while applying the color transformation. However, this method requires reference center's data that may not always be available. As a remedy, self-supervised learning can be employed that does not require GT [7]. It is based on first training the network on some pretext tasks and later utilizing for the other downstream tasks. In [19], a network is trained to carry out stain normalization by learning to map the color (stain) augmented images to the corresponding original images. This can be considered as self-supervised learning, wherein the pretext task is to learn to transform the color augmented images to the original images. Color transformation of test images to that of training data can be considered as the downstream task. This paper utilizes this approach but uses two coupled subnetworks (N1 and N2) instead of a single network and learns dual transformation (pretext tasks). In addition, classification heads are added to the encoders of both the subnetworks [5, 12]. The first subnetwork (N1) learns an identity transformation, while the second (N2) learns to perform stain normalization. N1 helps N2 learn the context-aware stain normalization task, while N2 helps N1 learn the stain-invariant features for the reconstruction task. Thus, both N1 and N2 assist each other, leading to improved stain-invariant performance on the subsequent classification task. We also experimented with segmentation as a downstream task. Since the proposed architecture can be utilized for stain normalization, classification, and segmentation, we name it an all-in-one network (AION). Elaborate experiments are presented on four datasets for each of the three tasks.

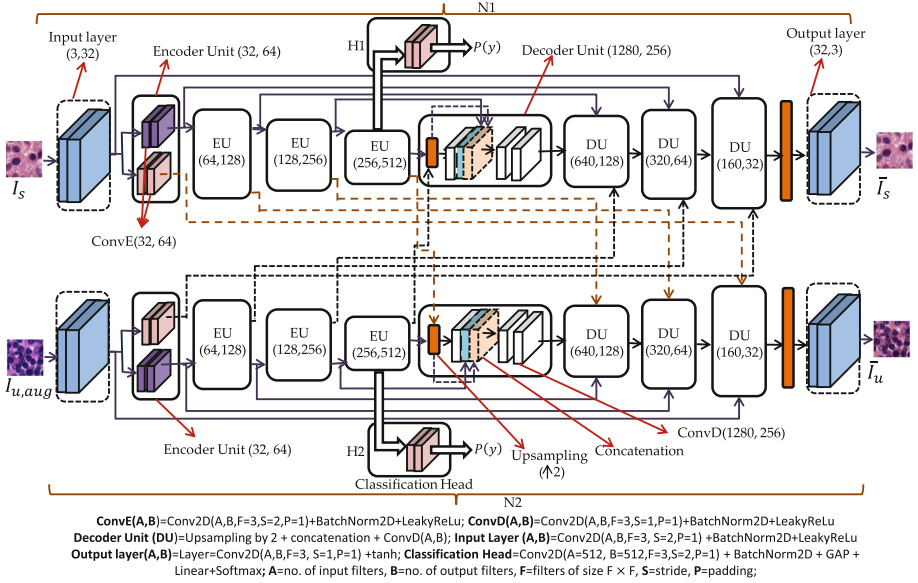


Fig. 1. AION: An architecture with two coupled-networks and classification heads for learning two transformations; one identity and another for stain normalization. AION can also be used for stain-invariant classification and segmentation.

2 Methods

Consider the source image dataset and the corresponding augmented dataset with color variations. The aim is to perform stain normalization such that the stain color profile of the augmented dataset is as close as possible to the source dataset. If trained successfully, the trained model can map one Center’s images to those of another Centre with a matching stain color profile. To accomplish this, we modify the architecture in [19] (here named as AION^{−−}) to arrive at AION, presented in Fig. 1.

Normalization by the AION Architecture: The AION architecture can be seen as a self-supervised learning architecture consisting of two parallel U-Net types AION^{−−} coupled through cross-connections running from the encoder units of one to the decoder units of the other. The first subnetwork, N1 is trained to learn the identity transformation on the source dataset. In contrast, the second subnetwork, N2 is supposed to map the augmented image dataset’s stain color profile to that of the source dataset. Each subnetwork consists of an input layer, four encoder units (EUs), four decoder units (DUs), and one output layer. Input layer and encoder units constitute the encoder, while decoder units and output layer constitute the decoder. The input layer consists of 32 filters, while the successive EUs consist of twice the filters than the preceding layer/unit. Hence, the first EU consists of 64 filters, and the fourth EU consists of 512 filters. The size of each filter is 3×3 . Each convolutional operation in the encoder is

performed with a stride of two and a padding of one. This leads to a reduction of output size by two after each convolutional operation. Each EU consists of two parallel convolutions layers and provides a coupling mechanism for N1 and N2. These parallel layers receive the same input from the preceding EU. However, the output from one goes to the succeeding EU of the same subnetwork and the output of the other goes to the decoder unit of another subnetwork. Similar cross-connections are followed for both the subnetworks (Fig. 1).

Each DU of N1 concatenates the input from the preceding layer/unit, the output from the corresponding EU of N1, and the output from the corresponding EU of N2. DUs also involve upsampling by two by the nearest neighbors approach to match the spatial size for concatenation. The concatenated output is given to the convolutional layers within DUs. The final output is forwarded to the next DU. Similar is the decoder structure of N2. Each DU contains half the number of filters of the preceding DU, with the first DU containing 256 filters and the fourth containing 32 filters. Finally, the output layer provides a three-channel output. Each convolutional operation in the network is followed by batch normalization (BatchNorm) and leaky ReLu except for the output layer, which has tanh activation and no BatchNorm.

Classification and Segmentation: To introduce the classification capability, we add two classification heads (CHs), one in each subnetwork on top of the last EU. Each CH consists of a single convolutional layer with 512 filters followed by BatchNorm, ReLu activation, global averaging pooling, and a classification layer. The CH of N1 predicts the class of images of the source dataset, while CH of N2 predicts the class of color augmented dataset. The AION can also be used for the segmentation. For this case, both N1 and N2 are fed with the source dataset, and the classification heads are not trained.

Training Methodology: To train the AION for stain normalization, an image I_s from the source dataset is given as input to N1. Another image $I_{u,aug}$, obtained through some transformation $\phi(\cdot)$ on image I_u from the source dataset, is given as input to N2, where indices u and s may or may not be equal. N1 aims to reconstruct the input image, while N2 attempts to match I_u . If we denote the output of N1 as \bar{I}_s and of N2 as \bar{I}_u , the loss function is given as:

$$\mathcal{L}_1 = MSE(\bar{I}_s, I_s) + MSE(\bar{I}_u, I_u), \quad (1)$$

where $MSE(\cdot)$ is the mean square error. Hence, N1 is learning an identity transformation, and N2 is learning ϕ^{-1} , assuming such inverse exists. Once trained, N2 should transform the stain (color) of the input data to that of the data input to N1. The classification heads are also trained simultaneously to predict the labels of the input images. Hence, the complete loss function is given as:

$$\mathcal{L} = MSE(\hat{I}_s, I_s) + MSE(\bar{I}_u, I_u) + NLL(y_1, \hat{y}_1) + NLL(y_2, \hat{y}_2), \quad (2)$$

where NLL is the negative log-likelihood, y is the true label, and \hat{y} is the predicted label by the classification head. The subscripts 1 and 2 denote the classification head of Network-1 and Network-2, respectively. *It should be noted that*

Table 1. Description of the four dataset. Highlighted cells for Camelyon17 represent the sum of train and val dataset of the respective columns as the architecture trained on one center’s is tested on another center’s whole data.

Application	Stain Normalization and Classification					Segmentation		
Datasets	Camelyon17 (Patches)					PCam (Patches)	Databowl (Patches)	CVCDB (Images)
	C0	C1	C2	C3	C4			
Train	53538	31169	54834	94635	16776	262146	8640	488
Val	6000	4000	4000	15002	4000	32770	1360	62
Test	59538	35169	58834	109637	20776	32770	2503	62

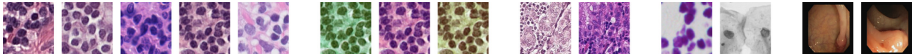


Fig. 2. Sample images from the datasets. (Set-1: left to right) Image from C0, C1, C2, C3, C4, and C5 of Camelyon17. (Set-2) augmented images from C3 obtained with transformation $\phi(\cdot)$. (Set-3) PCam. (Set-4) DSB, (Set-5) CVC.

classification Head-1 is trained on the source image data, while classification Head-2 is trained on the augmented data.

For segmentation, the tanh activations at the output layers are replaced with sigmoid activations. The input of both the networks is the same, and the classification heads are not trained. Also, the original (not augmented) data is used for training. The respective segmented mask is predicted from both the networks, and both the networks are trained simultaneously using the binary cross-entropy loss.

3 Results

In this work, we use four publicly available datasets. Camelyon17 challenge dataset [3] is used for stain normalization, PatchCamelyon (PCam) [20] for classification, and Data Science Bowl (DSB) [1] & CVC-ClinicDB (CVC) [2] for segmentation. Camelyon17 consists of hematoxylin and eosin (H&E) stained whole slide images (WSIs) of lymph node sections collected from five centers labeled as C_0 to C_4 . For training and testing, the patches of size 128×128 are extracted from the level-0 of the WSIs. PCam have 96×96 patches having a binary label indicating the presence or absence of metastatic tissue. There is stain-variation among the patches, which makes the classification task challenging. The DSB is a cell segmentation dataset having images of varying sizes. We have extracted 128×128 patches for training and testing. CVC is a polyps segmentation dataset with 612 images of size 384×288 pixels. For training with CVC, we have augmented the training set using random rotations. The distribution of the patches/images for all the datasets is provided in Table 1, and the sample images are shown in Fig. 2.

Stain-Normalization: For Camelyon17, there are variations in the colors of stained images of the five centers (Fig. 2). Intuitively, a classifier trained on one center’s data may perform unsatisfactorily on the remaining centers’ data. We train a classifier on a particular center’s data and test on the remaining centers’ data for stain-normalization. The stain normalization quality can then be assessed through improvement in the classification performance over the unnormalized data. We begin our analysis with C3 as the training data. For comparison, we chose Macenko [9] and Reinhard [13] as non-DL methods, and StainGAN [16] and AION⁻⁻ [19] as DL methods. StainGAN [16] is trained using C3 and C0 because it requires two training datasets. Results are also reported by adding a classification head to AION⁻⁻. We name this architecture as AION⁻⁻+H. Both AION and AION⁻⁻+H are trained with the same training methodology as discussed in Sect. 2. To obtain the augmented images, $\phi(\cdot)$ consists of saturation in the range [0, 2.5] and hue in the range [0.5,0.5] with a probability of 0.95. The sample augmented images are shown in Fig. 2. Some results are shown with an external classifier (EC) that consists of 10 convolutional layers, batch normalization and Leaky ReLU as activation, and a classification layer.

All the models are implemented using PyTorch 1.1.0 and trained using RTX2080 GPU. Except for classifiers, all the models are trained using Adam optimizer for 80 epochs, batch size of 32, and initial learning rate of 0.001, which is reduced to one-tenth of the present value if there is no change in the validation performance for seven epochs. For classifiers, an SGD optimizer and a batch size of 64 are used for 150 epochs. The initial learning rate of 0.001 is reduced to one-tenth at 80th, 120th, and 140th epochs. The following results are reported with AION⁻⁻+H and AION:

1. **AION⁻⁻+H (H):** Results from the classification head that is added to AION⁻⁻
2. **AION (H1):** Results with N1’s classification head of AION
3. **AION (H2):** Results with N2’s classification Head of AION
4. **AION⁻⁻+H (EC):** Results with an external classifier trained on a center, and tested on normalized images obtained from AION⁻⁻+H
5. **AION (N2+EC):** Results with an external classifier trained on a center, and tested on normalized images obtained from N2 of AION

Results with C3 as training data are shown in Table 2 in terms of balanced accuracy (BAC), the area under the curve (AUC), and the weighted F1-score (WF1). The mean performance of the methods requiring reference images is inferior to other methods, showing their dependency on the reference image’s choice. It can be seen that stain normalization leads to improved classification performance over original images for all the centers. For C0, AION (N2+EC) gives the top performance with a significant margin in terms of BAC and WF1. For C1 and C4, AION⁻⁻+H (EC) and AION (N2+EC) are having approximately similar performances, and for C2, AION⁻⁻ is giving the best results. Apart from C2, both AION⁻⁻+H (EC) and AION (N2+EC) are performing better than AION⁻⁻. This shows the contributions of the added components to the AION⁻⁻. Another significant observation is the better performance of AION

(H1) than original images. This shows that AION (H1) has also become stain-invariant, even though it is trained with the unnormalized images. This capability may have been introduced due to the coupling in the AION. AION⁻⁻+H (H) and AION (H2) are trained with augmented images and have seen varying stain colors. Hence, the performance improvement with these architectures is apparent. Table 3 shows the inter-center train-test results in terms of AUC. For all the centers, there is performance enhancement after stain normalization. Also, for most cases, results with AION are better than AION⁻⁻+H, which are better than AION⁻⁻.

Table 2. Results with C3 as training data, and other centers as the test data. The shaded cells represent mean results with five various reference images.

Dataset	C0			C1			C2			C4		
Metric/Method	AUC	BAC	WF1	AUC	BAC	WF1	AUC	BAC	WF1	AUC	BAC	WF1
Original	0.7835	0.5960	0.5422	0.7581	0.7013	0.6948	0.4026	0.3338	0.3187	0.8628	0.7826	0.7824
Macenko [9]	0.5530	0.5025	0.4539	0.6117	0.5410	0.4767	0.5499	0.5428	0.5203	0.6096	0.5907	0.5490
Reinhard [13]	0.5465	0.4898	0.3738	0.5833	0.5087	0.3791	0.4699	0.4759	0.2521	0.4353	0.4867	0.3532
StainGAN [16]	0.7926	0.5714	0.5023	0.8097	0.7217	0.7175	0.6882	0.6072	0.5241	0.5887	0.5097	0.365
AION ⁻⁻ [19]	0.8611	0.7192	0.7042	0.8257	0.7577	0.7553	0.9390	0.8067	0.7535	0.8115	0.7206	0.7137
AION ⁻⁻ +H (H)	0.7975	0.7199	0.7172	0.8327	0.7433	0.7398	0.7457	0.6751	0.6046	0.8739	0.8040	0.8039
AION ⁻⁻ +H (EC)	0.8438	0.7538	0.7497	0.8600	0.7805	0.7794	0.8174	0.7240	0.6567	0.9289	0.8661	0.8659
AION (H1)	0.7767	0.6941	0.6915	0.8040	0.7279	0.7248	0.6419	0.6072	0.6958	0.9286	0.8562	0.8562
AION (H2)	0.8019	0.7186	0.7161	0.8370	0.7395	0.7349	0.7833	0.6828	0.6008	0.8851	0.8129	0.8125
AION (N2+EC)	0.8661	0.7733	0.7712	0.8605	0.7781	0.7769	0.8579	0.7534	0.6937	0.9218	0.8627	0.8625

Table 3. AUC with inter-center training-testing

AUC								
Train	C0				C1			
Test	C1	C2	C3	C4	C0	C2	C3	C4
Original	0.8607	0.5076	0.7157	0.5080	0.8065	0.1852	0.6490	0.1052
AION ⁻⁻ [19]	0.8714	0.9341	0.9344	0.7714	0.7169	0.8880	0.7305	0.4088
AION ⁻⁻ +H (H)	0.8368	0.8915	0.9114	0.8973	0.7093	0.8448	0.8761	0.7907
AION ⁻⁻ +H (EC)	0.8183	0.9157	0.9304	0.9137	0.6674	0.8870	0.7787	0.4132
AION (H2)	0.8412	0.8956	0.9141	0.9029	0.7128	0.8465	0.8774	0.8221
AION (N2+EC)	0.8359	0.9130	0.9362	0.9206	0.6551	0.8680	0.8088	0.4920
Train	C2				C4			
Test	C0	C1	C3	C4	C0	C1	C2	C3
Original	0.6055	0.5722	0.3577	0.3617	0.4506	0.6532	0.7667	0.4012
AION ⁻⁻ [19]	0.6884	0.8269	0.7648	0.4211	0.5983	0.5797	0.8019	0.7130
AION ⁻⁻ +H (H)	0.6706	0.7722	0.7732	0.7732	0.5708	0.6920	0.7699	0.7466
AION ⁻⁻ +H (EC)	0.6772	0.8005	0.8455	0.5670	0.6125	0.6269	0.8264	0.7817
AION (H2)	0.6995	0.7800	0.7967	0.7940	0.5869	0.7121	0.7761	0.7529
AION (N2+EC)	0.6926	0.8034	0.8508	0.6623	0.6080	0.6418	0.7821	0.7576

Classification: To highlight AION’s classification capability, we use the binary class dataset of PCam with inter-image stain variations. We have also compared the results with AION⁻⁻+H to show the proposed architecture’s contribution. Results are summarized in Table 4.I. A methodology similar to stain normalization is used for training with PCam. First, we trained EC on the original PCam training set. This classifier provided an AUC of 0.8448 on the original PCam test set. The AION⁻⁻+H (H), AION (H1), and AION (H2) provide a gain of 4%, 4.61%, and 3.66%, respectively. Even though these classifiers are shallow (6 layers) than EC (10 layers), they provided a significant gain. This gain could be more with deeper heads. Hence, the proposed architecture can produce stain-invariant classifiers as a byproduct. To explore further, we stain-normalized PCam using AION⁻⁻+H and AION trained on C3 of Camelyon17. Training and testing with the resultant dataset provided an AUC of 0.8948 and 0.9105 with AION⁻⁻+H and AION (N2), respectively, which again proves AION’s usefulness. We also replaced the custom classifier with ResNet-24 and DenseNet-161. The best AUC is achieved with AION (N2) for both classifiers, which is 0.9356 with ResNet-34, and 0.9275 with DenseNet-161.

Table 4. (I) PCam Classification results, (II) CVC and DSB Segmentation results

(I) PCam Classification Results						
Classifier	Method	AUC		Classifier	Method	AUC
AION ⁻⁻ +H (H)	AION ⁻⁻ +H	0.8848		Resnet-34	Original	0.8948
AION (H1)	AION	0.8909			AION ⁻⁻ [19]	0.9059
AION (H2)	AION	0.8814			AION ⁻⁻ +H	0.9181
					AION (N2)	0.9356
EC	Original	0.8448		DensNet-161	Original	0.8691
	AION ⁻⁻ [19]	0.9054			AION ⁻⁻ [19]	0.9137
	AION ⁻⁻ +H	0.8948			AION ⁻⁻ +H	0.9091
	AION (N2)	0.9105			AION (N2)	0.9275
(II) Segmentation results on CVC and DSB with AION ⁻⁻ +H and AION						
Dice Similarity Coefficient (DSC)						
Dataset	CVC			DSB		
Initialization/Method	AION ⁻⁻ +H	AION (N1)	AION (N2)	AION ⁻⁻ +H	AION (N1)	AION (N2)
C3-Pretrained	0.6897	0.7453	0.6896	0.8670	0.8533	0.8682
Random	0.8243	0.8198	0.8328	0.8712	0.8723	0.8705
Intersection over Union (IoU)						
Dataset	CVC			DSB		
Initialization/Method	AION ⁻⁻ +H	AION (N1)	AION (N2)	AION ⁻⁻ +H	AION (N1)	AION (N2)
C3-Pretrained	0.7406	0.7765	0.7407	0.8527	0.8378	0.8542
Random	0.8361	0.8327	0.8432	0.8568	0.8579	0.8560

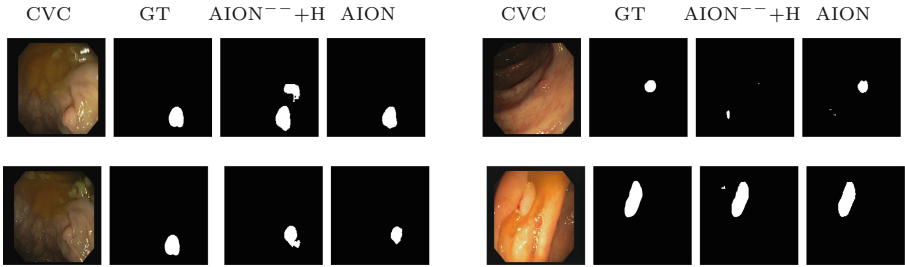


Fig. 3. Sample segmentation masks from AION⁺⁺+H and AION (H2) from CVC.

Segmentation: In one set of experiments, we considered segmentation as the downstream task and initialized AION⁺⁺+H and AION with the network’s weights trained on C3 (C3-Pretrained). In contrast, in the other experiment, they are initialized randomly. Also, as the classification head is not trained, AION⁺⁺ and AION⁺⁺+H are the same for segmentation. Both AION⁺⁺+H and AION gave an inferior performance with C3-pretrained networks as compared to random initialization (Table 4.II). This shows that stain-normalization-specific features are, perhaps, not helpful in the segmentation task on this dataset. Also, the performance difference between the two initializations is large for CVC as compared to DSB. This is due to the similarity in the imaging modality of DSB and C3. Also, AION has better performance than AION⁺⁺+H for both datasets. The qualitative results are also presented in Fig. 3.

4 Conclusion

In this work, we proposed a novel self-supervision based dual-transformation coupled-network architecture for stain normalization. The architecture is also equipped with classification heads that can achieve stain-invariant classification. The utility is also shown for the downstream task of segmentation. The architecture can perform well on all three applications of stain normalization, classification, and segmentation.

References

1. data science bowl. <https://www.kaggle.com/c/data-science-bowl-2018>. Accessed 5 Feb 2021
2. Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., Vilariño, F.: WM-DOVA maps for accurate polyp highlighting in colonoscopy: validation vs. saliency maps from physicians. *Comput. Med. Imaging Graph.* **43**, 99–111 (2015)
3. Bándi, P., et al.: From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Trans. Med. Imaging* **38**(2), 550–560 (2019)

4. Gupta, A., et al.: GCTI-SN: geometry-inspired chemical and tissue invariant stain normalization of microscopic medical images. *Med. Image Anal.* **65**, 101788 (2020)
5. Haghighi, F., Hosseinzadeh Taher, M.R., Zhou, Z., Gotway, M.B., Liang, J.: Learning semantics-enriched representation via self-discovery, self-classification, and self-restoration. In: Martel, A.L., et al. (eds.) *MICCAI 2020*. LNCS, vol. 12261, pp. 137–147. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59710-8_14
6. Jain, A.K.: *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc. (1989)
7. Jing, L., Tian, Y.: *Self-supervised visual feature learning with deep neural networks: a survey* (2019)
8. Kothari, S., et al.: Automatic batch-invariant color segmentation of histological cancer images. In: *From Nano to Macro, 2011 IEEE International Symposium on Biomedical Imaging*, pp. 657–660 (2011)
9. Macenko, M., et al.: A method for normalizing histology slides for quantitative analysis. In: *ISBI*, pp. 1107–1110 (2009)
10. Magee, D., et al.: Colour normalisation in digital histopathology images. In: *Proceedings Optical Tissue Image analysis in Microscopy, Histopathology and Endoscopy (MICCAI Workshop)*, vol. 100 (2009)
11. McCann, M.T., Majumdar, J., Peng, C., Castro, C.A., Kovačević, J.: Algorithm and benchmark dataset for stain separation in histology images. In: *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 3953–3957 (2014)
12. Mehta, S., Mercan, E., Bartlett, J., Weaver, D., Elmore, J.G., Shapiro, L.: Y-net: joint segmentation and classification for diagnosis of breast biopsy images. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) *MICCAI 2018*. LNCS, vol. 11071, pp. 893–901. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00934-2_99
13. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Comput. Graphics Appl.* **5**, 34–41 (2001)
14. Ruderman, D.L., Cronin, T.W., Chiao, C.C.: Statistics of cone responses to natural images: implications for visual coding. *JOSA A* **15**(8), 2036–2045 (1998)
15. Ruifrok, A., Ruifrok, D.: Quantification of histochemical staining by color deconvolution. *Anal. Quant. Cytol. Histol. /Int. Acad. Cytol. [and] Am. Soc. Cytol.* **23**(4), 291–299 (2001)
16. Shaban, M.T., Baur, C., Navab, N., Albarquni, S.: StainGAN: stain style transfer for digital histological images. arXiv preprint [arXiv:1804.01601](https://arxiv.org/abs/1804.01601) (2018)
17. Abe, T., Murakami, Y., Yamaguchi, M.: Color correction of pathological images based on dye amount quantification. *Opt. Rev.* **12**(4), 293–300 (2005)
18. Tabesh, A., et al.: Multifeature prostate cancer diagnosis and Gleason grading of histological images. *IEEE Trans. Med. Imaging* **26**(10), 1366–1378 (2007)
19. Tellez, D., et al.: Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Med. Image Anal.* **58**, 101544 (2019)
20. Veeling, B.S., Linmans, J., Winkens, J., Cohen, T., Welling, M.: Rotation equivariant CNNs for digital pathology (2018)
21. Zanjani, F.G., Zinger, S., Bejnordi, B.E., van der Laak, J.A.W.M.: Histopathology stain-color normalization using deep generative models. In: *1st Conference on Medical Imaging with Deep Learning (MIDL 2018)*, pp. 1–11 (2018)
22. Zanjani, F.G., Zinger, S., Bejnordi, B.E., van der Laak, J.A.W.M., de With, P.H.N.: Stain normalization of histopathology images using generative adversarial networks. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 573–577, April 2018. <https://doi.org/10.1109/ISBI.2018.8363641>