




Improving Accuracy of Nuclei Segmentation by Reducing Histological Image Variability

Yusuf H. Roohani^{1,2}✉  and Eric G. Kiss¹

¹ Stanford University School of Medicine,
291 Campus Drive, Stanford, CA 94305, USA
yusuf.x.roohani@gsk.com

² GlaxoSmithKline, 200 Cambridgepark Drive, Cambridge, MA 02140, USA

Abstract. Histological analyses of tissue biopsies is an essential component in the diagnosis of several diseases including cancer. In the past, evaluation of tissue samples was done manually, but to improve efficiency and ensure consistent quality, there has been a push to evaluate these algorithmically. One important task in histological analysis is the segmentation and evaluation of nuclei. Nuclear morphology is important to understand the grade and progression of disease. However, implementing automated methods at scale across histological datasets is challenging due to differences in stain, slide preparation and slide storage. This paper evaluates the impact of four stain normalization methods on the performance of nuclei segmentation algorithms. The goal is to highlight the critical role of stain normalization in improving the usability of learning-based models (such as convolutional neural networks (CNNs)) for this task. Using stain normalization, the baseline segmentation accuracy across distinct training and test datasets was improved by more than 50% of its base value as measured by the AUC and Recall. We believe this is the first study to perform a comparative analysis of four stain normalization approaches (histogram equalization, Reinhart, Macenko, spline mapping) on segmentation accuracy of CNNs.

Keywords: Histology · Stain normalization · Nuclei segmentation
Convolutional neural networks · Machine learning

1 Introduction

Diagnoses made by pathologists using tissue biopsy images are central for many tasks such as the detection of cancer and estimation of its current stage [2]. One routine yet important step within histological analyses is the segmentation of nuclei. Nuclear morphology is an important indicator of the grade of cancer and the stage of its progression [3]. It has also been shown to be a predictor of cancer outcome [4]. Currently, histological analysis such as these are done manually, with pathologists counting and evaluating cells by inspection. Developing

automated methods to perform this analysis will help pathologists maintain consistent quality, allow for greater use of histological analysis by reducing cost and throughput.

However, automating nuclei detection is not a trivial task and can be challenging for a number of reasons - one important challenge is the lack of stain standardization. Stain manufacturing and aging can lead to differences in applied color. It could also be the result of variation in tissue preparation (dye concentration, evenness of the cut, presence of foreign artifacts or damage to the tissue sample), stain reactivity or image acquisition (image compression artifacts, presence of digital noise, specific features of the slide scanner). Each stain has different absorption characteristics (sometimes overlapping) which impact the resulting slide color. Finally, storage of the slide samples can have aging effects that alter the color content [2, 5]. Radiologists have established standards (such as DICOM) to ensure consistency between scans from different origins and time. Ideally, histopathology would also work within a framework like DICOM where images can be standardized against experimental conditions to ensure consistency across datasets.

Recently, there has been considerable interest in the application of novel machine learning tools such as deep learning to aid in routine tasks such as segmentation. These models generally work on raw pixel values, but could achieve greater accuracy through reducing the variance contributed by slide and stain specific variables. However, the approach must not be too general or else false positives will occur through altering the image signal [1, 2].

The aim of this project is to address the impact of variability in histological images on the accuracy of deep learning based algorithms for segmentation. A Convolutional Neural Network (CNN) was trained to perform nuclei segmentation and tested to get a baseline. Four stain normalization techniques, histogram equalization, Reinhard, Macenko, and spline mapping were then applied as means to reduce color variability of the raw images. The CNN was trained and tested again for each of these normalization conditions to get segmentation accuracy in each case. This paper is unique in that it employs a wide variety of normalization methods, uses deep learning based nuclei segmentation accuracy as a metric, and tests the model on a different dataset to understand model generalizability.

2 Stain Color Normalization

Stain normalization techniques involve transforming image pixel values. There are a wide array of techniques in literature, but most involve statistical transformations of images in various color spaces. Below we provide an overview of the four techniques used. Since our goal was mainly to highlight the impact of stain normalization as opposed to finding the best approach, we acknowledge that there is scope for further expanding the following list (e.g.: Automatic Color Equalization [12], HSV channel shifting, adding random Gaussian noise etc.). We chose the following four because they were reasonably diverse and were easily

implementable using University of Warwick’s stain normalization toolbox [10]. For methods requiring stain vector estimation, this was done using an image from the training set in both cases of training and testing.

- **Histogram equalization:** Histogram equalization is a commonly used image processing technique that transforms one histogram by spreading out its distribution to increase image contrast. In this analysis, histogram equalization was performed on each RGB channel in Matlab, effectively normalizing the color intensities frequencies between two images [9].
- **Macenko color normalization:** The Macenko color normalization method transforms the images to a stain color space by estimating the stain vectors then normalizes the stain intensities. Quantifying the stain color vectors (the RGB contribution of each stain) provides a more robust means of manipulating color information [8].
- **Reinhard color normalization:** Reinhard color normalization aims to make one image ‘feel’ like another by transforming one image distribution to be closer to another. Reinhard transforms the target and source images into L* color space. This was created to minimize the correlation between channels. After the source and target images are in this colorspace, descriptive statistics are used to transform the target image’s colorspace as described in [7]. Finally, the average channel values of the source are added back to the data points and it is transformed back to RGB colorspace.
- **Spline mapping:** Conceptually, the spline mapping technique is similar to the Macenko technique in that it estimates the stain vectors, deconvolves the image, maps the stain intensity to a target image before reconstructing back in RGB colorspace. Khan makes contributions in automatic stain vector calculation using a classifier with global and local pixel value information, and a non-linear stain normalization method [5].

3 Image Segmentation Using CNNs

This section describes the methods used to generate deep learning based image segmentation models. The goal was to train a model using one dataset and to test using another. We aimed to perform this approach using different normalization strategies to narrow down on an approach that best reduced variability and improved performance.

3.1 Model Selection

We first trained and validated the model on the same dataset to make sure that our training procedure was working correctly. For this we used breast tissue slices from [3]. We randomly split the dataset into 70% train and 30% validation. After experimenting with different network architectures, the final architecture that was chosen after the validation procedure was (Conv-BNorm-ReLU)x6 - (Fully Convolutional) - Softmax [3].

We chose to use a fully convolutional network (FCN) instead of a regular fully connected network so as to enhance throughput and quickly return the network based on results [6]. This meant that, at the time of inference, our model could simply process an entire image in one pass instead of needing it to be broken down into pixelwise patches. However, for training the network, we used a fully connected ultimate layer, and fed patches instead of a whole image as input. This allowed us to have greater control over the size and composition of the training classes, given their skewed distribution in the training set. This is also why we chose FCN over more recent architectures that work better with whole images such as U-Net [13] and Mask-RCNN [14]. We also realize that there are deeper architectures that could be used with FCN for further improving pixel level accuracy but our main focus was on showing the value of stain normalization as opposed to finding the optimal architecture for segmentation. We used the Caffe deep learning framework to design these models.¹

3.2 Dataset

The training dataset consisted of 143 histological sections of breast tissue from the Case Western Reserve digital pathology dataset. Each RGB image was 2000×2000 pixels, $20\times$ magnification and was H&E stained. Manually annotated masks were provided for over 12000 nuclei. We found that, for training, a patch size of 64×64 (87.8% baseline validation accuracy) worked better for training than 32×32 (82% accuracy). A total of 400,000 unique patches were generated for each scenario.

However, it was not sufficient to randomly sample from non-nuclear regions as defined by the hand annotations. There was a significant probability of sampling unannotated nuclei while developing negative patches for the training set. To address this problem, we used the approach outlined by [3]. Nuclei are known to absorb greater levels of the eosin (red) stain and so the red channel in the images was enhanced. A negative mask was thus generated defining regions outside of these enhanced red zones that were deemed safe for non-nuclei class patch selection. We also made sure to allocated a third of the non-nuclei patches to boundaries around the nuclei so that these would be clearly demarcated in the output. Moreover, positive and negative samples were equal in number even after accounting for these changes. The model prediction accuracy was found to benefit from these approaches.

The test set was composed also of breast tissue slices from a hand annotated dataset provided by the BioImaging lab at UCSB [11]. Referred to as the BioSegmentation benchmark, these were 58 H&E stained images at a much smaller resolution (896×768). This dataset proved to be ideal for model testing because the images were quite different from our training set both in terms of image quality and resolution and also in terms of the staining used (more eosin content). Patching was not required for the test set because we were using a fully convolutional network.

¹ Code: <https://github.com/yhr91/NucleiSegmentation/tree/master/BMI-260>.

3.3 Training

Once our model architecture and dataset generation approach had been finalized, we began to train separate model for each of the normalization scenarios as shown in Fig. 1. We used a batch size of 1000 because that could fit comfortably in our memory (P100 GPU 16 GB \times 2).

There were four models - these corresponded to the four techniques outlined previously: Histogram Equalization (SH), Macenko (MM), Reinhard (RH), Spline Mapping (SM). There was also a model for the unnormalized (Unnorm) case. Model performance would generally begin to plateau around 5–10 epochs. We did not notice overfitting in any model until 25 epochs except in SM. However, we could not continue training much beyond that point due to time constraints.

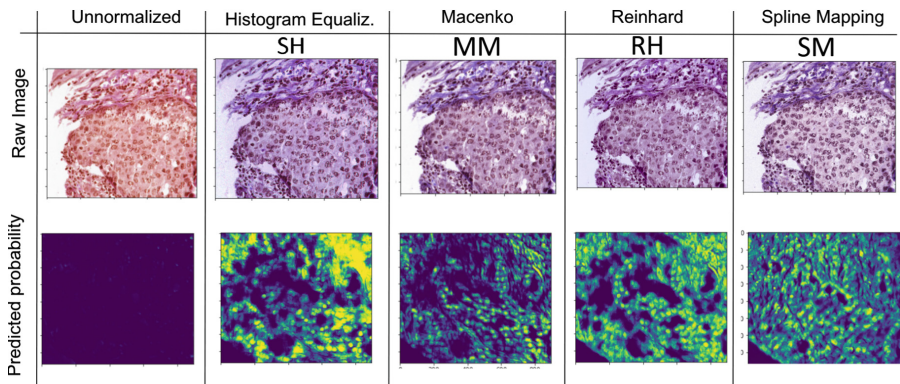


Fig. 1. The first row shows the test image as fed into the model after stain normalization (labelled using acronym). The normalization applied on the test image was the same as that applied on the training dataset in that case. The bottom row shows the model predicted output on the test images.

4 Results

4.1 Visual Inspection

The top row in Fig. 1 shows the original images after being transformed using the four different stain normalization approaches. We can see that all four images appear different in some respect. For example, HE and RH, which involve stain normalization through working directly with the color values show a noticeable blue tint. This is more pronounced in HE, where non-nuclear regions in the top right of the cell get quite heavily stained with hematoxylin. On the other hand, SM and MM, which both use stain vectors to map an input image to a target space, don't show a blue-ish tint and provide a much more robust transformation

that is true to the semantic information of the parent image (e.g.: treating nuclei and background regions distinctly).

The bottom row looks at the class probability for nuclear regions as predicted by models trained on datasets that were each stain normalized differently. Clearly all four normalized sets perform far better than the unnormalized dataset where almost no nuclei were detected due to the drastic change in staining as compared to what the model had been trained on. HE does pick up most of the nuclei but also a lot of background noise due to its inability to differentiate clearly between different types of staining. RH is also more sensitive to noise but does a better and clearer detection of nuclei as is visible in the clear boundaries. SM clearly performs the best at segmenting nuclei while also being most robust to false positives.

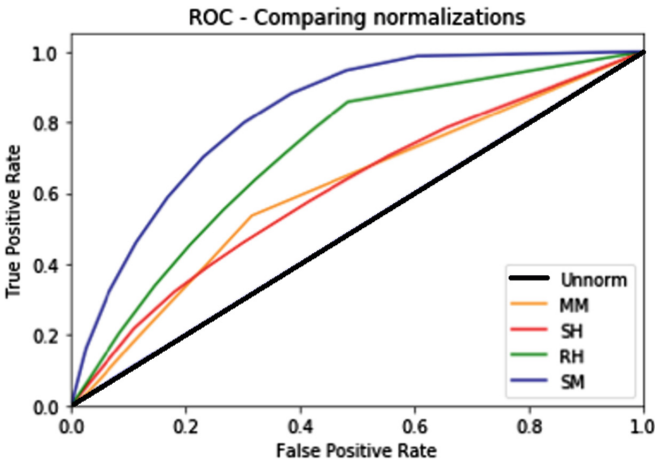


Fig. 2. ROC curve for models trained using different stain normalization schemes

4.2 Quantitative Assessment

To perform a more rigorous quantitative assessment, we looked at metrics calculated over a randomly selected set of 15 test images, using an output threshold of 0.5 for binarization (see Table 1). Simply calculating classification accuracy would be insufficient for this sort of segmentation problem. For instance, even if a classifier were to only classify pixels as non-nuclear regions it would still be around 85–90% accurate because the vast majority of pixels don't lie within nuclei.

Given the set of all nuclear pixels in the set, recall tells us what fraction of those were picked up by the model. Clearly SM does a great job in this area. SH and RH also do well but when we look at their precision values they are not as high as those for SM. Precision measures how many of the positives that you picked up were actually relevant. This indicates the tendency of SH and

RH to pick up more false positives than SM. This trade-off between true and false positives is best captured by the ROC curve (Fig. 2). Here, we see that the unnormalized case doesn't add any value at all while all the normalization scenarios show improved prediction accuracy. SM is the clear winner showing an excellent range of operation at a TPR of >80/90% while only allowing an FPR of 50%. This is very impressive considering how the model was trained on a staining visually very different from the one in the test data. This difference is quantitatively captured by the AUC. Finally, the F-score is another attempt to capture segmentation accuracy without getting bogged down by all the true negatives. It calculates the intersection of pixels that have been classified as nuclei in both the prediction and the ground truth and it divides that over the union of all pixels classified as nuclei by either set. Again, SM is seen to be the best at improving accuracy of the algorithm.

Table 1. Quantitative comparison of model performance under different forms of stain normalization

Normalization	Precision	Recall	F-score	AUC	Epochs
None	0.006	0.00	0.00	0.50	25
Histogram equalization (SH)	0.025	0.52	0.05	0.61	25
Macenko (MM)	0.026	0.18	0.05	0.61	25
Reinhard (RH)	0.04	0.55	0.07	0.71	12.5
Spline mapping (SM)	0.05	0.70	0.09	0.83	6

5 Discussion

Through this study, we have explored several stain normalization approaches that were all shown to reduce inter slide variability. The results (particularly AUC, F-score) clearly indicate that using a stain normalization approach increases the performance of the deep learning based segmentation algorithm. We found that SM performed better than all other approaches. We believe this is because it use a non-linear mapping function that is more accurate than the other approaches. It is able to delineate between different regions and map them appropriately to the target space.

We also noticed that the model seems to perform more poorly in case of normalizations that are biased more towards the eosin channel. In future, it may make sense to normalize the stain of the training dataset using two different approaches. This would push the model to become robust to these subtle changes and be less dependent on any one channel. Moreover, stain normalization could also be looked at as a regularization approach to enhance generalizability of deep learning based models in this space and prevent overfitting. On the other hand, we must remain conscious of the fact that staining color is a very valuable source of information in histological analyses and adopt a balanced approach towards stain normalization.

6 Conclusion

In this study, we looked at the impact of stain normalization as a means of improving the accuracy of segmentation algorithms across datasets. To the best of our knowledge, this is the first study that compares the chosen four stain normalization techniques through assessing their usability in the context of deep learning based segmentation models. There is scope for expanding upon this work with a deeper analysis of why certain normalization approaches or model architectures are better suited for this task.

References

1. Ghaznavi, F.: Digital imaging in pathology: whole-slide imaging and beyond. *Annu. Rev. Pathol.: Mech. Dis.* **8**, 331–359 (2013)
2. Irshad, H.: Methods for nuclei detection, segmentation, and classification in digital histopathology: a review’ current status and future potential. *IEEE Rev. Biomed. Eng.* **7**, 97–114 (2014)
3. Janowczyk, A., Madabhushi, A.: Deep learning for digital pathology image analysis: a comprehensive tutorial with selected use cases. *J. Pathol. Inform.* (2016)
4. Basavanthally, A., Feldman, M., Shih, N.: Multi-field-of-view strategy for image-based outcome prediction of multi-parametric estrogen receptor-positive breast cancer histopathology: comparison to oncotype DX. *J. Pathol. Inform.* **2**, S1 (2011). <https://doi.org/10.4103/2153-3539.92027>
5. Khan, K.M., et al.: A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution. *IEEE Trans. Biomed. Eng.* **61**(6), 1729–1738 (2014)
6. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015)
7. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Comput. Graph. Appl.* **21**(5), 34–41 (2001)
8. Macenko, M., et al.: A method for normalizing histology slides for quantitative analysis. In: *ISBI*, vol. 9, pp. 1107–1110, June 2009
9. https://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf. Accessed 24 Jul 2018
10. <http://www2.warwick.ac.uk/fac/sci/dcs/research/tia/software/sntoolbox>. Accessed 24 Jul 2018
11. <http://bioimage.ucsb.edu/research/bio-segmentation> . Accessed 24 Jul 2018
12. Rizzi, A., Gatta, C., Marini, D.: From retinex to automatic color equalization: issues in developing a new algorithm for unsupervised color equalization. *J. Electron. Imaging* **13**(1), 75–85 (2004)
13. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015*. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. IEEE, October 2017