Robust Color Normalization of Histopathological Whole Slide Images

BTP Project Report

Goutham Ramakrishnan, 140020039
Guide: Prof. Amit Sethi

November 24, 2017

Abstract

Deep learning techniques are extensively used today in computational pathology. However, they may be hampered by the variability in the appearances of the histological images. This necessitates the color normalization of these images. In their 2016 paper, Vahadane, Peng, Sethi et al. proposed a novel technique for stain separation and color normalization, based on sparse non-negative matrix factorization. This project implements the algorithm, enabling fast color normalization of very large whole-slide images, through robust patch sampling, efficient disk read-writes and a GPU-enabled implementation in TensorFlow. An improvement to the original technique, to prevent the tint induced in the white background of the normalized image, has also been proposed and implemented.

1 Introduction

Stained tissue samples are studied under the microscope in histopathology for disease diagnosis and prognosis. These images are analyzed using machine learning algorithms in computational pathology. However, there is often an undesired variability in the appearance of the images, due to differences in staining and scanning procedures. This hinders the performance of algorithms trained on these images and thus arises the need to normalize their appearances. Several color normalization schemes have been proposed for histological images. Normalization of large whole slide images(WSI) of tissues is difficult because of computational and memory constraints. This project overcomes these difficulties to enable color normalization on WSI's.

2 Background

Color Normalization of histological images is most commonly done by transforming the appearance of the images to match the appearance of a target image. It is important that the color normalization does not affect the underlying structure of the tissue, and modifies only the appearance. Color normalization techniques commonly used for natural images, such as histogram normalization tend to perform poorly on histological images. The color normalization scheme implemented in this project is based on a technique proposed by Vahadane, Peng, Sethi et al. Its advantages over other techniques have been discussed extensively in their research paper [1].

The most common reagent used for staining tissues is the combination of hematoxylin and eosin(H&E). This results in a distinction between the nuclei and the cytoplasm of the tissue, as they are primarily stained by the bluish-purple hematoxylin and red-pink eosin respectively. We can limit our discussion here to the H&E stain, however the topics discussed below are equally applicable to other staining schemes. A crucial step in several color normalization schemes is stain separation, i.e. the decomposition of an H&E image into its constituent stains. This is done by finding the color bases and the corresponding density maps of each stain. The individual stains can then be normalized independently, and combined to give the normalized image.

3 Methodology

Vahadane, Peng, Sethi et al.[1] present a novel technique for stain separation and color normalization. It shows better results than other techniques, both qualitative and quantitatively, as described in the paper. It utilizes several key insights obtained from by leveraging properties of histological images:

1. Non-negativity: The stain and optical densities of a given pixel in an image cannot be negative. Thus, non-negative matrix factorization(NMF) must be used to find the color basis and density matrices.
2. Sparsity: It is assumed that H&E reagent binds to specific the biological materials in a given ratio, and the material is therefore characterized by an *effective stain.* For example, the effective stains of all nuclei in an image is the same, whereas it differs from the effective stain of the cytoplasm.This assumption that each pixel contains only one stain allows us to impose a sparsity constraint on the NMF. By these assumptions, one no longer needs to annotate pure stains in the tissue. Using sparse non-negative matrix factorization(SNMF) ensures that the color bases are learnt in an unsupervised manner. Stain separation using SNMF does not extract pure H-stains and E-staínds, it extracts the effective H-stains and E-stains.
3. Soft-Classification: The sparsity constraint imposed can be controlled using a hyperparameter, which allows a small number of pixels to be constituted by more than one stain.

The motivation and reasoning behind each of the above assumptions is discussed in detail in [1].

3.1 Sparse Stain Separation

An overview of the stain separation technique is presented below.
Let:
m : number of color channels (m=3 for RGB images)
n : the number of pixels in the image
r : the number of stains (r=2 for H&E stained images)
I_0 : the maximum pixel intensity (I_0=255 for 8-bit images)

The relationship between the color basis matrix, color density matrix and the pixel intensity matrix is given by the Beer-Lambert Law.

$$I = I_0\, exp(-V) \quad ; \quad V = WH$$

I : $m \times n$ matrix of pixel intensities
V : $m \times n$ matrix of relative pixel intensities in the optical density space
W : $m \times r$ color basis matrix
H : $r \times n$ stain density matrix

The columns of W represent to the color basis vectors of each individual stain.
The rows of H represent the stain density vectors of each individual stain.

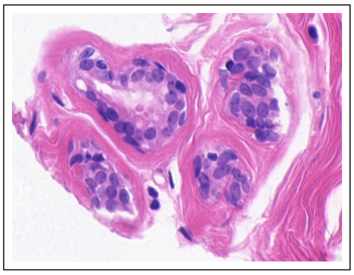

Figure 1: Original Image

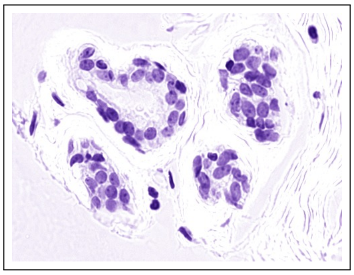

Figure 2: Effective H-stain

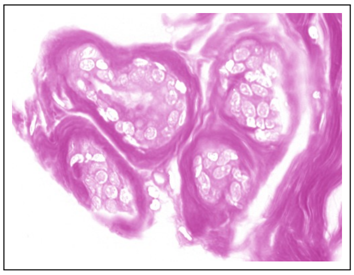

Figure 3: Effective E-Stain

The SNMF of the V matrix is achieved by solving the following optimization problem:

$$\min_{W,H} \|V - WH\|_F^2 + \lambda \sum_{j=1}^{r} \|H(j,:)\|_1 \ , \ W \geq 0, H \geq 0, \ \|W(:,j)\|_2^2 = 1$$

This is a modification of the NMF objective function. The sparsity constraint is introduced by adding the L1 norm of the rows of the stain density matrix. The sparsity is controlled by the value of λ. Uniqueness of the solution is enforced by constraining the color basis vectors to be unit norm.

The color basis matrix W is estimated through *Dictionary Learning*. The publicly available SPAMS library[3] was used for this purpose. The stain density matrix H can then be estimated using *sparse-coding* or by using the Moore–Penrose inverse of W.

An example of the structure preserving color normalization is shown in Figures 1-3.

3.2 Color Normalization

For structure preserving color normalization of a source with respect to a target, we reconstruct the source image using the color basis matrix of the target image. However, normalization of the source stain density vectors with respect to the target stain density vectors is also needed to ensure similarity in pixels intensities between the target and normalized image.

Overview of original algorithm:

1. Convert target image pixels into optical density space : $V_t = \log(\frac{I_0}{I_t})$
2. Estimate color basis and stain densities of target image : $V_t = W_t H_t$
3. Convert source image pixels into optical density space : $V_s = \log(\frac{I_0}{I_t})$
4. Estimate color basis and stain densities of source image : $V_s = W_s H_s$
5. Normalize source stain density matrix : $H_s^{norm} = \text{normalize}(H_s, H_t)$
6. Find normalized image in optical density space : $V_s^{norm} = W_t H_s^{norm}$
7. Reconstruct normalized image : $I_s^{norm} = I_0 \exp(-V_s^{norm})$

The normalize(H_s, H_t) function works by scaling the rows of the source stain density matrix by a constant. This ensures that the structure of the source image is preserved.
The function does the following:

$$H_s^{norm}(j:) = \frac{H_s(j,:)}{H_s^{RM}(j,:)} H_t^{RM}(j,:) \ , \ j = 1, \ldots, r.$$

where H^{RM} represents the robust pseudo-maximum of each row vector of H at 99%.

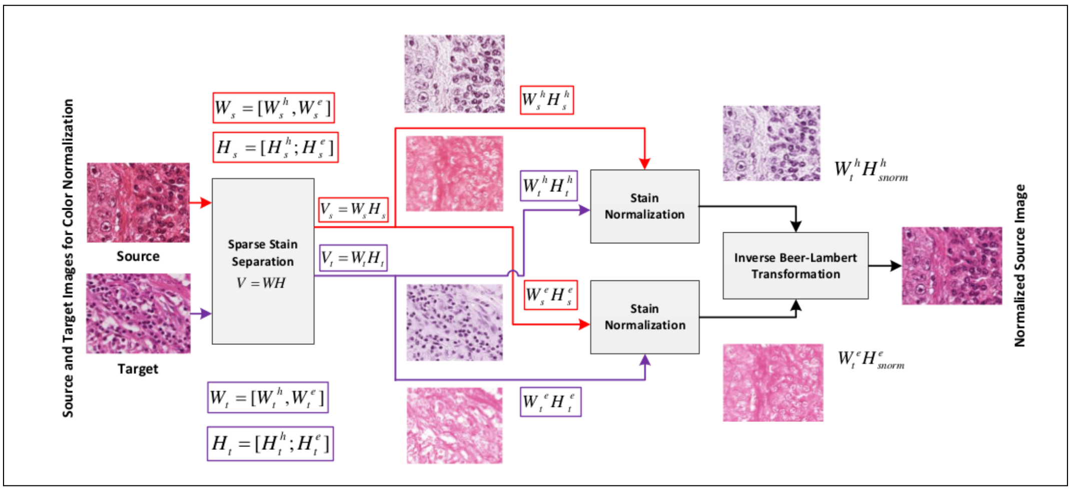

Figure 4: Overview of Color Normalization Algorithm (source:[1])

3.3 Challenges

1. Estimation of color basis for large images:
   The dictionary learning algorithm for estimation of W is computationally very expensive and becomes intractable for large images. A smart patch-based acceleration scheme was proposed in [1] to solve this problem. Suitable small patches are chosen from the large image, and W is individually estimated for each. The final W matrix of the large image is estimated by calculating the element-wise median of the W matrices of the patches, and normalizing the columns to have unit norm. This estimation scheme results in significant speed-up of the algorithm for large images.

2. Tinted whitespace in normalized image:
   Histopathological images are often scanned with a white background. When normalized, the whitespace in the image is often affected by a tint (pinkish or bluish, depending upon the dominant stain in the target image). This undesirable tint can be prevented by making a modification to the Beer-Lambert transform step in the algorithm.
   The BL transform assumes that the background maximum intensity of the image is equal to the maximum intensity possible and that it is a constant across all color channels. This is not always true. In the improved algorithm, we estimate the maximum intensities in each color channel, by sampling a large number of white pixels from the image. These values are then used to individually find the BL-transforms of each color channel. The final normalized image is then reconstructed using the maximum intensity values of the target image. This procedure necessitates modifications to steps 1,3 and 6 of the above algorithm.

   Modifications to original algorithm:

   1. Convert target image pixels into optical density space : $V_{t,C} = \log(\frac{I_{0,t,C}}{I_{t,C}}) \,\forall\, C$
   3. Convert source image pixels into optical density space : $V_{s,C} = \log(\frac{I_{0,s,C}}{I_{s,C}}) \,\forall\, C$
   6. Find normalized image in optical density space : $V_{s,C}^{norm} = W_{t,C} H_s^{norm} \,\forall\, C$

   C represents the color channels(R,G & B). $I_{0,t,C}$ and $I_{0,s,C}$ are the estimated maximum intensities of the C channel in the target and source images respectively. The V matrix is assembled row by row(channel by channel). $W_{t,C}$ and $W_{s,C}$ represent the rows corresponding to channel C of the target and source image respectively. Note that it is necessary to set the pixel intensities which are greater than the chosen threshold for a given color channel in the I matrices to be equal to the threshold (before steps 1 and 3).

4 Implementation

Some key implementation details of the algorithm in this project have been listed below:

1. Packages used:
   The implementation has been done entirely in python[5]. Two versions of the code exist, one which uses only NumPy[6] and a more efficient version which uses TensorFlow[2]. The SPAMS toolbox[3] is used for its dictionary learning functionality. The OpenSlide library[4] is used to read whole slide images. Several other python packages have also been used such as sklearn, cv2 and multiprocessing.

2. Whole Slide Image Compatibility:
   The following features were implemented to enable color normalization of large images:

   (a) *Patch-wise matrix computations:* Normalization of large images all at once requires very large memory capacity. H-estimation and reconstruction of the normalized image was done one longitudinal patch at a time. This procedure needs disk read-writes.

   (b) *Robust Patch Sampling:* It is necessary to find a sufficient number of valid patches for estimation of the color basis matrix. This is often a difficult process, especially for whole slide images with a lot of whitespace. The implementation is robust to the whitespace in the image, and recursively samples random patches from the image to ensure a good number of valid patches for reliable W estimation.

(c) *Maximum intensity estimation:* Sampling of a large number of white pixels from the image was done in parallel with the patch sampling procedure for efficiency. For each color channel, the 80th percentile of pixel intensities was used as a heuristic to estimate the maximum intensities of each channel. Note that this is the key step for the improved algorithm described above.

(d) *Percentile approximation:* Finding the 99th percentile of the entire H-matrix is computationally intractable when the number of pixels is very large. As an approximation, the 99th percentile of each longitudinal patch processed was computed, and their median was used for normalization of the H-matrix.

3. Optimizations:
   The following features implemented ensure fast and efficient color normalization:

   (a) *Parallel Processing:* The color basis estimation of the sampled patches is done in parallel on available cores, to maximize efficiency.

   (b) *GPU-enabled:* The TensorFlow implementation enables optimal usage of available computational resources.

   (c) *Normalization of image batches:* Functionality to normalize a batch of source images using the same target image is made available.

4. Other Points to note:

   (a) *Hyperparameters:* There are several hyperparameters in the implementation, which can be tuned as per need. Some of these parameters include the sparsity constant(λ), patchsize, number of patches, etc.

   (b) *RAM usage:* There are parameters in the code which can be adjusted to optimally use the available RAM.

   (c) *GPU usage:* TensorFlow often uses all available GPU's. Care must be taken to avoid conflicts between competing processes, to prevent program crashes.

   (d) *Color Basis estimation failure:* The estimation of the W matrix might fail in some situations. Care must be taken to ensure that the elements of any column of the W matrix are not uniformly $0.577(\frac{1}{\sqrt{3}})$. This situation may arise when the image has very little of any given stain. The stain separation demo must be run on the image to check if it is working properly. For some images, it is necessary to estimate W using the entire image rather than patches for proper stain separation.

5 Results

Efficient Color Normalization:
The project has enabled efficient and fast color normalization of whole slide images. The approximate time taken for processing images of different sizes is shown in Table-1.

Image Size(px)	Time taken
2000x2000	$<$ 30 s
5000x5000	$<$ 30 s
10000x10000	$<$ 1 min
20000x20000	3-4 min
40000x40000	5-7 min
80000x80000	25-35 min

The above simulations were run using an 6GB Nvidia GPU and HDD. It is to be noted that the times listed in the tables are estimates. The actual time taken for an image depends on the structure of the image as well as the computational resources available.

For smaller images, the W estimation step is most time consuming. For larger images, the computation time is heavily influenced by the disk read-writes necessary. Despite this, note that the time taken for larger images scales fairly well. Using SSD drives speeds up the procedure tremendously, reducing the time taken to normalize the 40000x40000 image to less than 3 minutes.

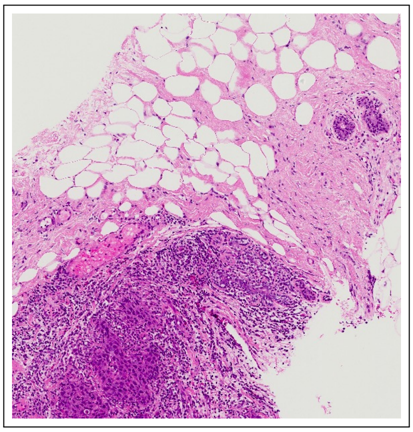

Figure 5: Source Image

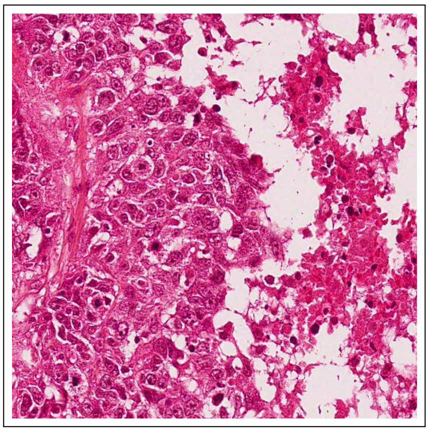

Figure 6: Target Image

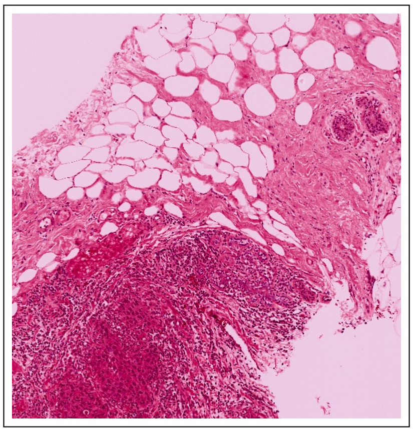

Figure 7: Original normalization

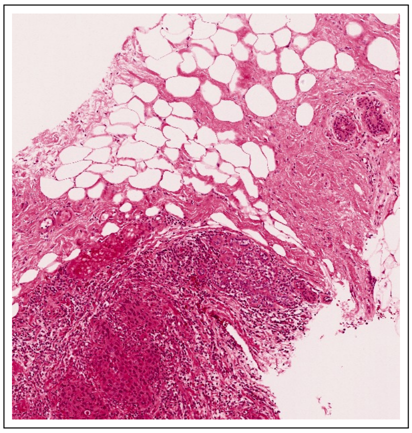

Figure 8: Improved normalization

Improved Normalized Image Appearance:
A comparison of the original and the improved color normalization algorithms is shown in the above figures. Figure 5 was normalized using Figure 6 as the target image. Figure 7 and Figure 8 show the outputs of the original and improved algorithms respectively.

The white background acquires a red tint when the original algorithm is used as seen in Figure 7. The improved algorithm fixes this drawback, as seen in Figure 8. The white background in this figure is similar in appearance to the white background in the target image used. It is to be noted that the choice of target image in this case is not optimal. The normalization demonstrated above is purely for the illustration of the improved normalization algorithm.

6 Conclusion

Color Normalization is a ubiquitous procedure in computational pathology. The structure preserving color normalization algorithm implemented in this project presents a superior alternative to the most contemporary widely used techniques. The functionality for direct normalization of WSI's also eliminates the need to extract patches from the image for individual normalization, before using them as input for deep learning networks.

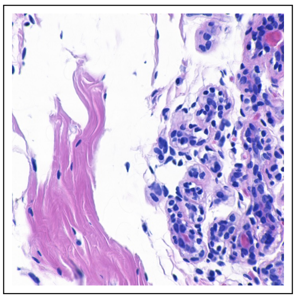

Figure 9: A good choice of target image

To achieve good color normalization, it is essential to use a suitable target image. An ideal target image for H&E stained tissues should contain a good amount of both hematoxylin and eosin, and also some whitespace. Good choices for target would be images such as Figure 1 and 9. It is also recommended to use small images as target, instead of whole slide images. The quality of the normalization is unpredictable when using whole slide images as target.

We plan to submit the improved color normalization technique as a one-page paper to the IEEE International Symposium on Biomedical Imaging (ISBI) 2018. The project implementation will be made public on github. It is anticipated that the technique and the implementation will gain popularity in the computational pathology community.

References

[1] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposito, Nassir Navab. *Structure-Preserving Color Normalization and Sparse Stain Separation for Histological Images.* IEEE Transactions on Medical Imaging, Vol. 35, No. 8, August 2016.

[2] TensorFlow Documentation.
`https://www.tensorflow.org/api_docs/`

[3] SPAMS: a SPArse Modeling Software
`http://spams-devel.gforge.inria.fr/doc-python/html/index.html`

[4] OpenSlide Python
`http://openslide.org/api/python/`

[5] Python 2.7.14 Documentation
`https://docs.python.org/2/index.html`

[6] NumPy Documentation
`https://docs.scipy.org/doc/numpy-1.13.0/reference/`