Deep Learning for Breast Cancer Mitosis Detection

Mike Dusenberry Fei Hu

Center for Open-Source Data & AI Technologies (CODAIT)
IBM
San Francisco, CA

mwdusenb@us.ibm.com

May 31, 2018

1 Introduction

Breast cancer is the leading cause of cancerous death in women in less-developed countries, and is the second leading cause of cancerous deaths in developed countries, accounting for 29% of all cancers in women within the U.S. [1, 2]. Increased survival rates have been attributed to early detection and improved treatment [2], giving incentive for pathologists and the medical world at large to develop improved methods for even earlier detection. The rate at which cancer cells proliferate is a strong indicator of a breast cancer patient's prognosis, and this measure is one of three components in the modified Bloom & Richardson grading system for invasive breast cancer [3]. The most common technique for determining the proliferation speed is through mitotic count estimates, in which a pathologist counts the dividing cell nuclei in hematoxylin and eosin (H&E)-stained slide preparations to determine the number of mitotic bodies per high power field (HPF). Unfortunately, this approach is known to have reproducibility problems due to subjectivity in counting [4], giving rise to the need for improved, more objective approaches.

Much recent work has been done on the task of automated mitosis detection and other related tasks [5, 6, 7, 8, 9, 10]. In this paper, we present a deep learning approach to the task of automated mitosis detection. Our main contributions are the inclusion of color augmentation, minority class oversampling, noise marginalization at prediction time, a custom residual network (ResNet) model, and a more efficient prediction algorithm. Additionally, we have released the codebase[1].

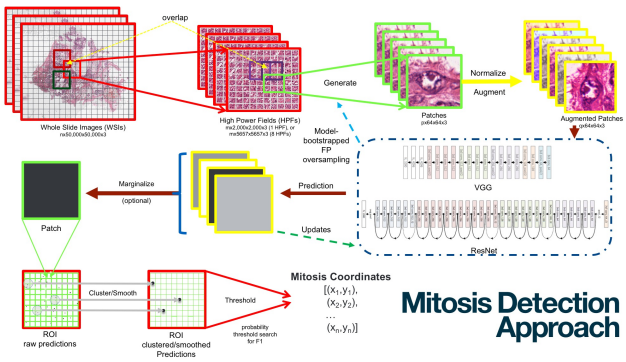

Figure 1: Mitosis detection approach.

2 Methods

At a high level, as shown in Figure 1, our approach begins by preprocessing a dataset of regions of tissue from whole slide images (WSIs) of breast tumors into a dataset of mitotic and non-mitotic patches. We then train a convolutional neural network (CNN) model to predict the presence of a mitotic figure in a given patch. Given an initial trained model, we preprocess the raw dataset again with model-based false-positive (FP) oversampling to generate a more difficult training dataset. We then train a new model on this second dataset. To make predictions on a new image region, we apply the model to the image in a sliding window fashion with noise marginalization, yielding a prediction at each location. A clustering algorithm is then used to smooth the potentially noisy set of predictions into a set of final predictions for mitosis locations.

Preprocessing We begin with a dataset of breast cancer cases, in which each case contains images of regions of tissue extracted from WSIs of breast tumors

[1] `https://github.com/CODAIT/deep-histopath`

and accompanying exhaustive lists of (x, y) coordinates corresponding to the center of each mitotic figure. The dataset is randomly split case-wise into training and validation sets, stratified by lab of origin, if available, and constrained to include some percentage of the cases in the training set. We then generate training and validation datasets of mitotic and non-mitotic square image patches in which each side is of length l_{prep}. Mitotic patches are extracted from locations centered at the given labeled coordinates. Non-mitotic patches are extracted in a sliding window fashion with a stride s_{prep}, and are constrained to be at least $2d$ pixels in Euclidean distance apart from the center of each mitotic figure. Additionally, we sample from the set of all potential non-mitotic patches by drawing, for each patch, a decision from a Bernoulli distribution governed by a parameter p.

Training Given our preprocessed data, our training algorithm proceeds as follows. First, we randomly shuffle the training set, augment and normalize each example, and generate batches of size n_{train} examples by sampling from the training set, possibly with oversampling of the minority mitosis class. Our data augmentation algorithm includes a random rotation; a random translation via random cropping to a square with sides of length l_{train} such that the center is within $d - 5$ pixels in Euclidean distance from the original center; random flips along the spatial axes; and color augmentation instead of color normalization [11]. Our color augmentation approach uses parameterized random brightness, contrast, saturation, and hue adjustments, with values based on recent work [9]. We normalize each example to the interval $[-1, 1]$. Minority oversampling via class-aware sampling [12] is used to yield balanced batches, and we define an epoch as the number of batches needed to exhaust the list of non-mitotic patches. Minority oversampling has been shown to be an effective approach to improve classification performance of CNNs in class-imbalanced problems [13].

Our first model is a modified ResNet [14], specifically a ResNet-50, with pretrained weights[2]. We replace the final affine and softmax functions with a new binary affine function and a sigmoid function, update the stride on the first convolution layer from 2×2 to 1×1 for "height $\times$ width", and update the final average pooling function to a kernel size of 4×4.

Our second model is a custom ResNet using the updated residual block formulation [15] with bottlenecks. The model contains an input block, three groups or "stages" of three residual blocks each, and an output block. The input block has a single 2D convolution function with a kernel of shape $(3, 3, 3, 16)$ for (height, width, input depth, output depth), 2×2 stride, and appropriate spatial padding such that the output tensor has the same height and width as the input tensor, i.e., "same" padding. The first "stage" is defined as a set of three residual blocks with 3×3 kernels and 1×1 strides, where the output kernel depth is 32. The second and third stages are each defined as a residual block with a 3×3 kernel and 2×2 stride, followed by two residual blocks with 3×3 kernels and 1×1 strides. The output kernel depth of the second stage is 64, and that of the third stage is 128. The output block has a batch normalization function [16], a ReLU function [17], a global average pooling function [18], an affine function to yield a single output value per example, and a sigmoid function. All convolution function parameters are initialized by sampling from a standard Gaussian distribution scaled by $\sqrt{\frac{2}{D}}$ [19], where D is the number of values in a single example of the input tensor for the given function, and the final affine function parameters are initialized by sampling from a standard Gaussian distribution scaled by $\sqrt{\frac{1}{D}}$ [20]. All batch normalization functions use a momentum of 0.9 and an epsilon term of 1e$-$4.

We use the logistic loss function, assuming a Bernoulli distribution parameterized by the model for each prediction. Additionally, we add an L2 regularization term to the loss with a regularization parameter λ, assuming a standard Gaussian prior on all model parameters. We employ transfer learning to train the first model by first optimizing the loss with respect to the parameters of the new affine layer for e_1 epochs using the Adam optimizer [21] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon =$ 1e$-$8, and a learning rate lr_1. Following that, we optimize the loss with respect to all model parameters for e_2 epochs using stochastic gradient descent with Nesterov momentum [22], with momentum μ, and learning rate lr_2. We train the second model by optimizing the loss with respect to all model parameters for e epochs using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon =$ 1e$-$8, and a learning rate lr.

Training proceeds by optimizing the loss for some number of epochs e. At the end of each epoch, we evaluate the model on the training and validation preprocessed datasets, and save a checkpoint. Evaluation metrics include positive predictive value (PPV) (i.e., precision), sensitivity (i.e., recall), and maximum F1 score. The best model is selected as the checkpoint that has the largest maximum F1 score on the preprocessed validation dataset.

False-positive Oversampling The best ResNet-50 model trained on the initial preprocessed training dataset will exhibit a poor F1 score due to a low pre-

[2] `https://www.tensorflow.org/versions/master/api_docs/python/tf/keras/applications/ResNet50`

cision. Motivated by work by Cireşan et al. [5], we use the initial best model to generate a new intermediate preprocessed training dataset composed only of FP cases. A FP case is defined as a candidate non-mitotic patch that is predicted as mitotic by the model. Candidate non-mitotic patches are sampled following the preprocessing procedure outlined on page 1. These non-mitotic FP patches are then combined with the original preprocessed training dataset to yield a new model-bootstrapped preprocessed training dataset in which FP cases have been oversampled. The validation dataset remains the same to allow for proper comparisons. Collectively, this new dataset is used to train the custom ResNet model.

Prediction In order to make predictions of the locations of mitotic figures on the original raw dataset, we proceed as follows. We first apply the best trained model to the original images in a sliding window fashion with a stride s_{pred}. At each location, we either record the output of the model applied to a centered patch, or we apply noise marginalization and record the ensuing output. Noise marginalization acts as a complement to data augmentation, where the latter injects noise η during the training process to model $p(y|x,\eta)$, and the former marginalizes over the noise at test time to model $p(y|x) = \int p(y|x,\eta)d\eta$. In practice, for each location we form a batch of n_{pred} data-augmented variants of the centered patch, apply the model to this batch, and average over the batch of predictions to yield a single noise-marginalized prediction for the given location.

Evaluation of location predictions proceeds by defining a true-positive (TP) as a ground-truth mitosis for which there exists at least one prediction within d pixels in Euclidean distance from the mitosis; a FP as any prediction that is greater than d pixels from the closest mitosis; and a false-negative (FN) as any ground-truth mitosis for which there exists no predictions within d pixels from the mitosis. The F1 score is then computed across all predictions.

Depending on the stride, multiple predictions could fall within the d pixel radius from a mitosis location. Although multiple predictions near a mitosis will count as a single TP, each prediction that is not near a mitosis will count as a separate FP. Thus, in order to reduce the potential for FPs, we propose a clustering algorithm based on the DBSCAN algorithm [23] in which we cluster the predictions above a probability threshold t_1, and then extract the cluster centers as the predictions. For clustering, we introduce a variable ϵ as the maximum distance between two samples for them to be considered as part of the same cluster, and a variable m as the minimum number of samples for a candidate cluster to be considered a valid cluster. Finally, we extract from these predictions those with a probability greater than a threshold t_2 as the final predictions.

	Training	Validation
Mitoses	1206	346
Non-mitoses	189305	960444

Table 1: Numbers and types of patches in the original preprocessed dataset.

Hyperparameter Optimization We use random grid search [24] to optimize the hyperparameters for each stage of the approach.

3 Experiments & Results

Datasets The main mitosis detection dataset[3] contains 73 cases from three different pathology centers in the Netherlands. Of the 73 cases, 23 of them are represented by regions of interest ranging in area from 7 mm^2 to 58 mm^2 (median 26 mm^2), each cut up into separate images that each represent a single HPF of size 0.5×0.5 mm^2. The other 50 cases are each represented by a single image region of size 2 mm^2. All slides were stained with H&E, and were scanned at 40x magnification with a spatial resolution of 0.25 µm/pixel. The images were exported in the TIFF format. For each image, all mitotic figures present were annotated by recording coordinates corresponding to the center of each mitosis, and these annotations were saved in CSV files. All annotations required the consensus of at least two pathologists. An additional 34 cases with the same details as the latter 50 training cases are provided as a testing dataset on which to make final predictions.

In our original preprocessed dataset, 80% of the cases were placed in the training set, $l_{\text{prep}} = 135$ pixels, $s_{\text{prep}} = 4$ for the training set, $s_{\text{prep}} = 32$ for the validation set, $p = 0.001$ for the training set, and $p = 1$ for the validation set. We set $d = 30$ pixels, which represents 7.5 µm. Table 1 shows the number and types of patches found in our original preprocessed dataset.

We generated FP cases for the training set cases using $s_{\text{prep}} = 8$ and all other settings mentioned previously. In addition, we also generated FP cases from the ICPR 2012[4] and ICPR 2014[5] mitosis datasets, using $s_{\text{prep}} = 4$ for the former, and $s_{\text{prep}} = 32$ for the latter. Table 2 shows the number and types of patches in our FP-oversampled dataset.

Experiments Table 3 describes the hyperparameter search space and final values for training the first model

[3] http://tupac.tue-image.nl/node/3
[4] http://ludo17.free.fr/mitos_2012/dataset.html
[5] https://mitos-atypia-14.grand-challenge.org/dataset

	Training	Validation
Mitoses	3160	346
Non-mitoses	305316	960444

Table 2: Numbers and types of patches in the FP-oversampled preprocessed dataset.

Hyperparameter	Search Space	Best Values
n_{train}	32, 64, 128	64
l_{train}	64	64
λ	0 – 1e−2	6.92e−3
e_1	5	5
lr_1	1e−5 – 1e−2	4.74e−3
e_2	10	10
lr_2	1e−7 – 1e−2	4.02e−3
μ	0.85 – 0.95	0.913
oversample	true, false	true

Table 3: Hyperparameter search space and final values for training the first model on the original preprocessed dataset. Comma-separated values indicate discrete options, while en-dashes indicate intervals.

on the original preprocessed dataset. Table 4 describes the hyperparameter search space and final values for training the second model on the FP-oversampled preprocessed dataset. Table 5 describes the hyperparameter search space and final values for the full prediction task on the original validation dataset using the best model.

Tables 6, 7, 8, and 9 describe the evaluation results on the original preprocessed dataset, FP-oversampled dataset without marginalization, FP-oversampled dataset with marginalization, and original dataset, respectively. The results on the preprocessed datasets serve as a proxy to the final task of predicting mitosis locations for the original dataset given a full HPF rather than a single patch.

Hyperparameter	Search Space	Best Values
n_{train}	32, 64, 128	128
l_{train}	64	64
λ	1e−7 – 1e−4	2.51e−5
e	100	100
lr	1e−5 – 1e−3	7.71e−4
oversample	true, false	true

Table 4: Hyperparameter search space and final values for training the second model on the FP-oversampled preprocessed dataset. Comma-separated values indicate discrete options, while en-dashes indicate intervals.

Hyperparameter	Search Space	Best Values
s_{pred}	1, 16, 32	16
marginalization	true, false	true
n_{pred}	128	128
t_1	0.0 – 1.0	0.4
t_2	0.0 – 1.0	0.59
ϵ	60	60
m	1, 2, 3	2

Table 5: Hyperparameter search space and final values for making predictions on the original validation dataset. Comma-separated values indicate discrete options, while en-dashes indicate intervals.

Metric	Train	Validation
F1	0.9836	0.4078
PPV	0.9815	0.07410
Sensitivity	0.9858	0.8873

Table 6: Intermediate results on the original preprocessed dataset.

Metric	Train	Validation
F1	0.9649	0.6495
PPV	0.9571	0.3385
Sensitivity	0.9727	0.8179

Table 7: Intermediate results on the FP-oversampled preprocessed dataset without marginalization. Note that this validation set is equivalent to the original preprocessed validation set.

Metric	Train	Validation
F1	-	0.7194
PPV	-	0.4177
Sensitivity	-	0.8584

Table 8: Intermediate results on the FP-oversampled preprocessed dataset with marginalization. Due to computational complexity, only the validation set is evaluated. Note that this validation set is equivalent to the original preprocessed validation set.

Metric	Train	Validation	Test
F1	-	0.6041	**0.6012**
PPV	-	0.6131	**0.6084**
Sensitivity	-	0.5954	**0.5941**

Table 9: Final results for the full prediction task on the original dataset.

References

[1] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2018. *CA: a cancer journal for clinicians*, 68(1):7–30, 2018.

[2] Lindsey A Torre, Freddie Bray, Rebecca L Siegel, Jacques Ferlay, Joannie Lortet-Tieulent, and Ahmedin Jemal. Global cancer statistics, 2012. *CA: a cancer journal for clinicians*, 65(2):87–108, 2015.

[3] Khawla Al-Kuraya, Peter Schraml, Joachim Torhorst, Coya Tapia, Boriana Zaharieva, Hedvika Novotny, Hanspeter Spichtin, Robert Maurer, Martina Mirlacher, Ossi Köchli, et al. Prognostic relevance of gene amplifications and coamplifications in breast cancer. *Cancer research*, 64(23): 8534–8540, 2004.

[4] Mitko Veta, Paul J Van Diest, Mehdi Jiwa, Shaimaa Al-Janabi, and Josien PW Pluim. Mitosis counting in breast cancer: Object-level interobserver agreement and comparison to an automatic method. *PloS one*, 11(8):e0161286, 2016.

[5] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks. In *MICCAI*, pages 411–418, Berlin, Heidelberg, September 2013. Springer, Berlin, Heidelberg.

[6] Kyunghyun Paeng, Sangheum Hwang, Sunggyun Park, Minsoo Kim, and Seokhwi Kim. A Unified Framework for Tumor Proliferation Score Prediction in Breast Histopathology. December 2016.

[7] Maxime W Lafarge, Josien P W Pluim, Koen A J Eppenhof, Pim Moeskops, and Mitko Veta. Domain-adversarial neural networks to address the appearance variability of histopathology images. *arXiv.org*, July 2017.

[8] Andrew Janowczyk and Anant Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, 7(1):29, 2016.

[9] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, Jason D Hipp, Lily Peng, and Martin C Stumpe. Detecting Cancer Metastases on Gigapixel Pathology Images. *arXiv.org*, March 2017.

[10] Mehmet Günhan Ertosun and Daniel L Rubin. Automated Grading of Gliomas using Deep Learning in Digital Pathology Images: A modular approach with ensemble of convolutional neural networks. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2015:1899–1908, 2015.

[11] Marc Macenko, Marc Niethammer, J S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A Method for Normalizing Histology Slides for Quantitative Analysis. *ISBI*, pages 1107–1110, 2009.

[12] Li Shen, Zhouchen Lin, and Qingming Huang. Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks. *ECCV*, 2016.

[13] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *arxiv.org*, October 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv.org*, December 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. *arXiv.org*, March 2016.

[16] S Ioffe and C Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[17] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conference on Machine Learning*, 2010.

[18] Min Lin, Qiang Chen, and Shuicheng Yan. Network In Network. *arXiv.org*, December 2013.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun 0001. Delving Deep into Rectifiers - Surpassing Human-Level Performance on ImageNet Classification. In *International Conference on Computer Vision*, 2015.

[20] Yann LeCun, Leon Bottou, Genevieve B Orr, and Klaus Robert Müller. Efficient BackProp. In *Neural Networks: Tricks of the Trade*, pages 9–50. Springer, Berlin, Heidelberg, Berlin, Heidelberg, 1998.

[21] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[22] Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[24] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 2012.