Semi-Supervised Learning for Cancer Detection of Lymph Node Metastases

Amit Kumar Jaiswal*[1], Ivan Panshin*[2], Dimitrij Shulkin*[3], Nagender Aneja[4], and Samuel Abramov[5]

[1]University of Bedfordshire [2]Perm State University [3]Schaeffler Group [4]Universiti Brunei Darussalam
[5]Abramov Software GmbH & Co. KG

Abstract

Pathologists find tedious to examine the status of the sentinel lymph node on a large number of pathological scans. The examination process of such lymph node which encompasses metastasized cancer cells is histopathologically organized. However, the task of finding metastatic tissues is gradual which is often challenging. In this work, we present our deep convolutional neural network based model validated on PatchCamelyon (PCam) benchmark dataset for fundamental machine learning research in histopathology diagnosis. We find that our proposed model trained with a semi-supervised learning approach by using pseudo labels on PCam-level significantly leads to better performances to strong CNN baseline on the AUC metric.

1. Introduction

The scale of digitization of pathology scans is still moderate, and recent research has proclaimed heterogeneity and disagreement amongst pathologists diagnoses [32]. However, high-resolution digitization of microscopic images has inspired computer vision researchers to work in the field of pathology diagnosis. The digitization of whole-slide images (WSI) from glass slides has stimulated researchers to implement state-of-the-art convolutional neural networks (CNNs) in medical imaging. A CNN trained on patches extracted from WSI serves to recognize metastatic cancer detection. CNNs has been exhibited to perform better than pathologists in several tasks. This is partly due to the success of ImageNet 2012 challenge[1] and also due to the adaptability of CNNs to medical imaging applications. CNNs comprising of different layers of nodes are essentially pattern recognizers. This property of CNNs has been exploited in medical imaging. A CNN trained on a set of images that have been split into patches correctly labeled by well qualified and medical practitioners can differentiate different parts of an image. A trained CNN network can accept an input of unlabeled image to predict if there is a cancer tumor cell or not. Some studies have compared the performance of pathologists with algorithmic results showing some algorithms performed better in terms of accuracy and time efficiency [2, 5]. Liu et al. [19] implemented CNN on Camelyon16[2] dataset for lesion-level tumor detection and achieved above 97% AUC score in comparison to 73.2% sensitivity achieved by a human pathologist. Additionally, the approach found that two slides in the training set erroneously labeled.

Further, in recent years deep convolutional neural networks (DCNNs) have improved significantly in the area of computer vision including image recognition and have been widely applied and accepted to enhance healthcare facilities. Litjens et al. [17] classified digital pathology and microscopy techniques in three broad categories namely (1) detection, segmentation, and classification of nuclei, (2) segmentation of a large organ, and (3) detection and classification of a disease. Computerized digital pathology techniques have improved due to the introduction of challenges in pathology. Annotated whole-slide images provided in Camelyon16 challenge allowed participants to use deep learning models such as VGG, ResNet, and GoogLeNet. Top solutions used one of these architectures.

In this work, we present a semi-supervised learning approach that outperforms, even more, the performance of CNN [33] in terms of the AUC metric. Our proposed DenseNet based model is evaluated on a slightly modified version of the PCam dataset. The original PCam dataset contains duplicate images due to its probabilistic sampling, however, our evaluation follows the same dataset with no duplicates in it. Otherwise, the same data and splits as the PCam benchmark dataset are maintained.

Our paper is organized into six sections. Having introduced the extent of the paper in Section 1 followed by semi-supervised learning approach in Section 2 which in-

*These authors are the corresponding authors and contribute equally to this study

[1] http://www.image-net.org/challenges/LSVRC/2012/

[2] https://camelyon16.grand-challenge.org/Data/

cludes the problem formulation, model architecture and the algorithm applied in effectuating the training steps for better performance of the prediction result (the tumor labels). Next, we will discuss the adapted new distribution on PatchCamelyon benchmark dataset including the techniques applied during training steps. In Section 4, we will report the evaluation results followed by related work in Section 5, while Section 6 concludes the paper.

2. Semi-Supervised Learning

We use a semi-supervised learning approach for incremental training of our proposed model to leverage the unlabeled instances for achieving learning performance. Below we formulate the problem and describe our algorithmic approach for detecting metastatic cancer.

2.1. Problem Settings

The cancer detection task is a binary image classification problem, where the input is a small (96 x 96px) digital histopathology image *I* and the output is a binary label $l \in \{0, 1\}$ stipulating the absence or presence of metastases in small image patches respectively.

Every single sample in the training set, we optimize the binary cross entropy loss[3]

$$B_L(I, l) = -l \log p(Y = 1|I) - (1 - l) \log p(Y = 0|I)$$

where $p(Y = i|I)$ refers to the probability that the network specifies to the label i.

2.2. Model Architecture

For this identification task, we use DenseNet which is a classic CNN architecture that was created [10] in order to solve the vanishing gradient problem [23]. Unlike other architectures that address this issue, like ResNets [8] or highway networks [29], whereas in DenseNet all layers are connected so that the information flow between layers in the network is maximal (Figure 1). In other words, such connectivity pattern introduces $\frac{L(L+1)}{2}$ connections in an L-layer network. Figure 1 illustrates this layout schematically.

To be more precise, the proposed DenseNet201 model uses compression of 0.5 with no bottleneck layers. In other words, if a dense block contains *m* feature-maps, the following transition layer generates $\lfloor 0.5m \rfloor$ output feature-maps.

Moreover, after removing the top layers and instead of fully connected layers, we concatenated the global average pooling (GAP) and global max pooling (GMP) layers including batch normalization (BN) layer. Also, we use dropout layer (0.6) with a dense layer having one output which includes sigmoid activation.

We concatenated the GAP and GMP layers to use as a slight modification of a strategy described in [15]. In this paper, it is proposed to replace the traditional, fully interconnected layers in CNN by GAP. The idea is to create a feature map for each corresponding category of the classification task. Instead of placing fully connected layers over the feature maps, one should take the mean and max of each feature map, and the resulting vector is fed directly after BN and dropout layers into the sigmoid plane. One advantage of GAP and GMP layers across the fully interconnected layers is that it is more native to the convolutional structure by forcing correspondences between feature maps and categories. Another advantage is that there is no parameter for optimization in GAP and GMP layers, which avoids overfitting at this level.

By placing the dropout layer after BN the strategy described in [14] was followed. Past work [28] introduced dropout as an easy way to prevent CNNs from overfitting. It has been proven significantly effective in a variety of machine learning areas such as image classification [31]. Before the birth of BN, it became a necessity for almost all modern networks and successfully increased their performance against overlay risks despite their amazing simplicity. Past work [11] demonstrated BN, a powerful capability that not only accelerated all modern architectures but also improved their strong baselines through their role as regularizers. Therefore, earlier work has employed BN in almost all current network structures [31, 9, 38] and proves its high practicability and effectiveness.

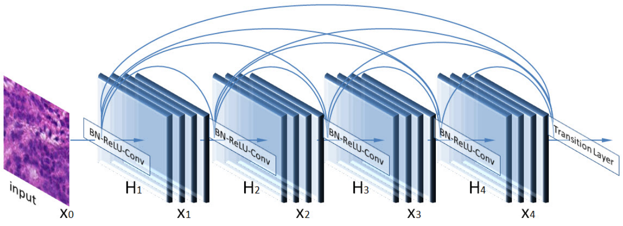

Figure 1. **DenseNet201 Block Architecture**

2.3. One Cycle Policy

In this work, we use one cycle policy approach. It was first introduced for SGD [26]. One cycle policy is a slight modification of cyclical learning rate policy (CLR) where a minimum and maximum learning rate limits with a step size was specified [24]. This policy allows the loss to plateau before the training ends. It combines the advantages of curriculum learning [3] and simulated annealing [1], both of which have a long history of use in deep learning.

As shown in Figure 2 the step size is the number of iterations used for each step, and a cycle consists of two such steps - one in which the learning rate (LR) increases and the other in which it decreases. With one cycle policy, the cycle is always smaller than the total number of iterations where the learning rate descends several orders of magnitude less than the initial learning rate for the remaining iterations.

[3] https://keras.io/losses/#binary_crossentropy

The maximum learning rate of 0.00055 led to the best results. For selecting the minimum learning rate, we divided the maximum learning rate by a factor of 10.

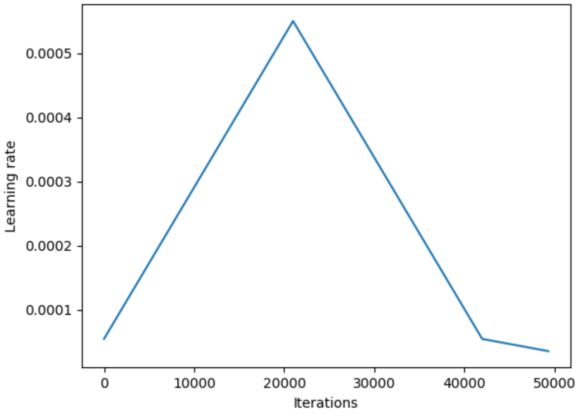

Figure 2. **One Cyclic Policy - Learning Rate**

Momentum and learning rate are closely related. The optimal learning rate depends on the momentum and the momentum depends on the learning rate [25]. Also, they found in their experiments that cyclical momentum led to better results. In practice, they recommend choosing two values such as 0.85 and 0.95 and reducing them from the higher to the lower value when the learning rate increases, then returning to the higher momentum when the learning rate decreases.

2.4. Pseudo Labels in Cancer Identification

In this section, a semi-supervised learning approach as described in [13] is applied where we train a convolutional neural network several times in a supervised manner with labeled and unlabeled data simultaneously. For unlabeled data, pseudo-labels that include the class with the maximum predicted probability are used as if they were real labels.

This method is actually equivalent to entropy regularization [7] where the conditional entropy of class probabilities can be used for a measure of class overlap. By minimizing entropy for unlabeled data, the overlap of the class probability distribution can be reduced. It promotes differentiation between low-density classes, which is often assumed in semi-supervised learning.

Considering multi-layer neural networks with M layers of hidden units it generally follows for the output unit h_i of kth layer:

$$h_i^k = s^k \left(\sum_{j=1}^{d^k} W_{i_j}^k h_j^{k-1} + b^k \right), k = 1, ..., M+1$$

where s^k is a non-linear activation function of the kth layer, $W_{i_j}^k$ is the weight of kth layer connecting input unit j with output unit i, h_j^{k-1} is the input value of previous layer, b_i^k is the bias factor of kth layer corresponding to output unit i, $f_i = h_i^{M+1}$ are output units used for prediction of target class and $x_j = h_j^0$ are input values. Since this is a binary classification problem, the sigmoid function is used for the output representing probability of true positive label.

The global network is to be trained by minimizing supervised loss function:

$$\sum_{i=1}^{C} L(y_i, f_i(x))$$

where C is the number of labels, y_i is the 1-of-K code of the label, f_i is the network output for ith label and x is the input vector. Since sigmoid is used for the output for the cancer classification task, cross entropy is given as:

$$L(y_i, f_i(x)) = -y_i log f_i - (1 - y_i) log(1 - f_i)$$

Incorporating pseudo labels into overall loss function gives the following expression:

$$L = \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{C} L(y_i^m, f_i^m) + \alpha(t) \frac{1}{n'} \sum_{m=1}^{n'} \sum_{i=1}^{C} L(y_i'^m, f_i'^m)$$

where n is the number of mini-batch in labeled data for stochastic gradient descent (SGD), n' for unlabeled data, f_i^m is the output units of m's sample in labeled data, y_i^m is the label of that, $f_i'^m$ for unlabeled data, $y_i'^m$ is the pseudo-label of that for unlabeled data and $\alpha(t)$ is a coefficient balancing them.

Entropy regularization [7] is a way to benefit from unlabeled data within the maximum a posteriori estimate. This scheme allows separating low-density classes without modeling the density by minimizing the conditional entropy of class probabilities for unlabeled data:

$$H(y|x') = \sum_{m=1}^{n'} \sum_{i=1}^{C} P(y_i^m = 1|x'^m) \log P(y_i^m = 1|x'^m)$$

where n' is the number of unlabeled data, C is the number of classes, y_i^m is the unknown label of the mth unlabeled sample and, x'^m is the input vector of mth unlabeled sample. The entropy is a measure of class overlap. As class overlap decreases, it lowers the density of data points at the decision boundary. The mean average precision (MAP) estimate is defined as the maximizer of the posterior distribution:

$$C(\Theta, \lambda) = \sum_{m=1}^{n} \log P((y^m | x^m; \Theta) - (y|x'; \Theta))$$

where n is the number of labeled data, x^m is the mth labeled sample, λ is a coefficient balancing two terms. By maximizing the conditional log-likelihood of labeled data (the first term) with minimizing the entropy of unlabeled data (the second term), one can get the better performance using unlabeled data.

Pseudo-labels are target classes for unlabeled data as if they were real labels. For the first training run, only the labeled data was used. From the second fine-tuning training run, the following assumption was made:

$$y_i' = \begin{cases} 1 & \text{if } P(TP) > 0.9 \\ 0 & \text{if } P(TN) < 0.1 \end{cases}$$

where $P(TP)$ is the predicted probability of a true positive label and $P(TN)$ is the predicted probability of a true negative label. With this assumption, fine-tuning training runs were repeated five times. After each fine-tuning training run, it is possible to increase the pseudo label set until a certain convergence is achieved.

Because the total number of labeled data and unlabeled data is quite different and the training balance between them is quite important for the network performance, pseudo labels with the ratio of 1:1 to the training and validation set were added considering the balance between assumed true positive and true negative pseudo labels. With this approach, it was possible to increase the area under the curve (AUC) of DenseNet201 model after ten fine-tuning training runs.

3. Adapting New PatchCamelyon Data

We use the *PatchCamelyon*, a comprehensive patch-level data set derived from Camelyon16 data. In this context, a new benchmark is developed that can accommodate the high volume, quality, and diversity of Camelyon16. The PCam dataset contains 327680 patches extracted from Camelyon16 at a size of 96 x 96 pixels with 10 magnification, selected using a hard negative mining regime. Since metrics at slide-level potentially obscure the relative performance of patch-level models. It has been proposed earlier [33] to validate them on a patch-level task. Through this dataset, the task of histopathology diagnosis becomes accessible as a challenging benchmark for fundamental machine learning research. Based on the PCam dataset presents results that are consistently better than results of Camelyon16 state-of-the-art approaches, including [19, 34].

We analyse the 49% of test data in the first phase, we found no issue with the distribution of various histopathological scans. As the training data is huge with labeled and unlabeled data. However, simply training a neural network on the PCam dataset to predict labels turns out yielding very poor results. To address this issue, proper distribution of targets in a test data is needed and we employed a slightly modified version[4] of the original PCam dataset for this work.

3.1. Training

For cancer detection task, we trained our model for ten times with new re-prediction of pseudo labels after each training run, where each training run consists of seven epochs. Before training, we removed 498 images with too many white and black pixels which contain no structure information as outliers from the training set to reduce noise which is illustrated in Figure 3.

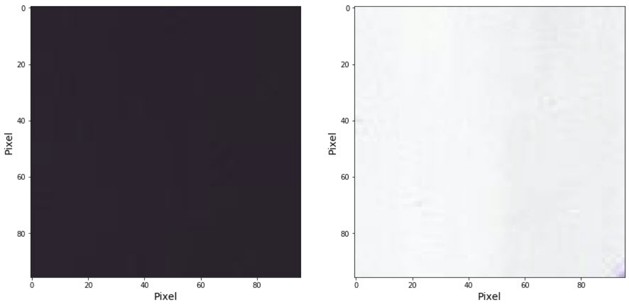

Figure 3. **Images as Outliers in the Train Set**

Finally, we resize the images from 96 x 96 to 224 x 224 pixel as the pre-trained models were originally trained on this size. After each semi-supervised learning run, more and more pseudo labels could be predicted, thus the training corpus could be increased where we perform random split to train and validation set.

Moreover, we apply a set of 10 online data augmentations. We describe the particular transformations in Section 3.2.

3.2. Test Time Augmentation

We apply test-time augmentation (TTA) during testing. TTA is a powerful technique that refers to performing data augmentation on a test image in order to get several versions of it and average predictions for them.

Test-time augmentation has been shown to improve the performances of computer vision algorithms [36]. Typically transformations include flipping, cropping, rotating, scaling, etc. Having applied a set of transformations during test-time include ten transformations: horizontal flip, vertical flip, rotation from -45 to +45 degrees, cropping each side by 0-20%, scaling by 80-120%, translation from -20% to +20% (per axis), sharpening and overlying the results with the original using an alpha between 0.0 and 1.0, embossing and overlying the results with the original using an alpha between 0.0 and 1.0, Gaussian noise, changing hue and saturation. In other words, for each original image in test set there are ten modified versions. The model makes

[4] https://www.kaggle.com/c/histopathologic-cancer-detection/data

eleven predictions and these predictions are blended with equal weights giving the final prediction for the image.

The transformations for TTA are identical to the ones used in cross-validation. This is done so that cross-validation can be used as a reliable metric of how well an algorithm performs on unseen data (test set).

4. Evaluation

4.1. Results

We apply semi-supervised learning approach on different pre-trained models[5] such as VGG16, InceptionResNetV2, InceptionV3, Xception, ResNet101 and DenseNet201 in the sense of transfer learning.

Table 1. **Evaluation Results**

Model	51% Test Data	49% Test Data	100% Test Data
VGG16	0.9768	0.9721	0.9745
InceptionResNetV2	0.9764	0.9769	0.9766
Xception	0.9748	0.9756	0.9752
InceptionV3	0.9758	0.9790	0.9774
SE-ResNet101	0.9784	0.9781	0.9783
DenseNet201	**0.9786**	**0.9802**	**0.9794**
GDenseNet [33]			0.9630

As shown in the Table 1 the DenseNet201 model performs better than other deep CNN models. This is illustrated in 4.

The characteristic for all pre-trained models was the fact that they were already over-fitted after 5-7 epochs. The Figure 5 shows the losses of train and validation sets including pseudo labels during the 10th fine-tuning run of DenseNet201 model. When considering the validation loss, a sign of overfitting after the 5th epoch can be detected, as the validation loss begins to increase.

4.2. Ensembles

Ensemble methods can help to reduce variance [12] and improve the overall performance of machine learning algorithms [4, 20, 18].

For this cancer detection task, we obtained the best result with the ensembling technique. We train several versions of SE-ResNet101 with an extensive TTA after the predictions from all models by averaging with equal weights. Such approach provided the best and the most robust results.

The biggest downside of such an approach is that it's computationally expensive. We trained 7 SE-ResNet101 for this ensemble. Each model took 6 hours to train and another hour to get predictions using NVIDIA Tesla P100.

We evaluated TTA with a different set of transformations (compared to transformations for the DenseNet model). We use 15 transformations: vertical flip, horizontal flip, rotations by 90, 180, 270 degrees, horizontal flip and rotation by 90 degrees, horizontal flip and rotation by 270 degrees, changing brightness, contrast, saturation, hue, and also 3 different combinations of changing brightness, contrast, saturation, and hue.

Table 2. **Comparison of the best single model and the ensemble**

Model	51% Test Data	49% Test Data	100% Test Data
Ensemble (7 SE-ResNet101)	0.9810	0.9822	0.9816
Best single model (DenseNet201)	0.9786	0.9802	0.9794
GDenseNet [33]			0.9630

As shown in Table 2, the ensemble technique led to the AUC of 0.9816 (evaluated on 100% of test data) outperforming the best single model as well as the benchmark solution presented in [33].

5. Related Work

Veeling et al. [33] proposed rotation equivariant CNNs showing that rotation equivariance improved tumor detection on a challenging lymph node metastases dataset. The authors suggested a fully-convolutional patch-classification model that is equivariant to 90” rotations and reflection. The model has shown a notable advance on the Camelyon16 benchmark [2] dataset.

Bejnordi et al. [2] assessed the performance of automated deep learning algorithms at identifying metastases in hematoxylin and eosinstained tissue regions of lymph nodes of women with breast cancer and compared it with pathologists diagnoses in a diagnostic setting. The experiments results revealed that some deep learning algorithms succeeded more excellent diagnostic performance than a panel of 11 pathologists competing in a simulation study intended to mimic regular pathology workflow; algorithm performance was comparable with a specialist pathologist interpreting whole-slide images without time constraints.

Gorelick et al. [6] implemented a two-stage AdaBoost-based classification for automatic prostate cancer detection and grading on hematoxylin and eosin-stained tissue images. The first stage named tissue component classification includes automatic tessellation of an image into superpixels utilizing a graph-cut based approach; extraction of superpixel appearance, morphometric and geometric features; and classification of superpixels in nine tissue component types based on the extracted features using modest AdaBoost. In the second stage, the authors classified cancer versus non-cancer and low-grade versus high-grade cancer utilizing tissue component labeling. The approach produced a 60-times reduction in data size and thus increasing processing efficiency the results have shown 90% accuracy for cancer versus non-cancer and 85% for high-grade versus low-grade classification. The false-negative rate was 12% for cancer detection and 5% for high-grade cancer detection.

Sun et al. [37] implemented deep learning algorithms for lung cancer diagnosis on lung image database consortium

[5] We experimented with all other models as mentioned and comparing it in Table 1

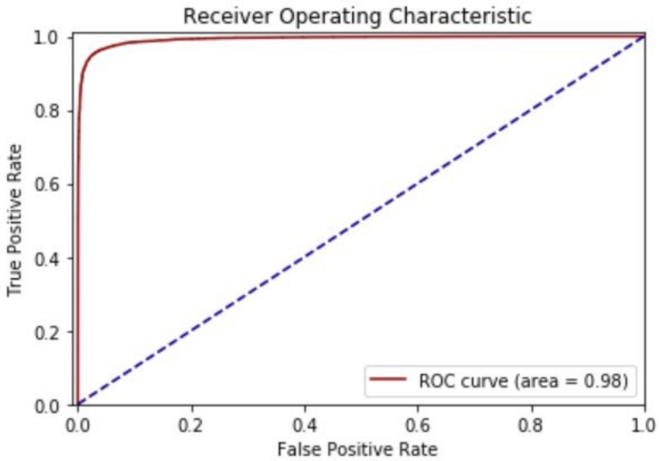

Figure 4. **Area under the ROC Curve**

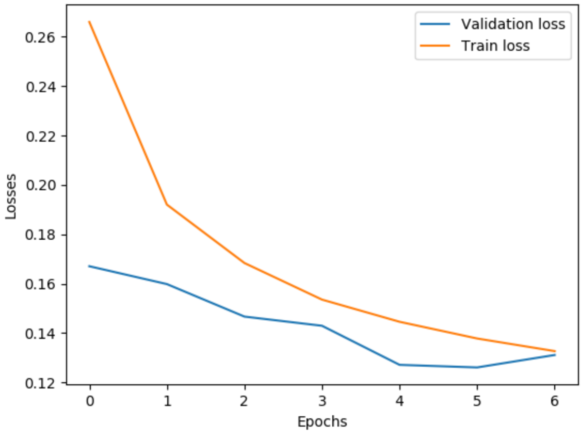

Figure 5. **Validation and Train Loss**

(LIDC) database. The authors implemented a convolutional neural network, deep-belief network (DBN), stacked denoising autoencoder (SDAE). CNN architecture comprises eight hidden layers with odd-numbered convolutional layer and even-numbered pooling and sub-sampling. Each convolutional layer employed 12, 8, 6 feature maps and connected to pooling layers with the 5 x 5 kernel. The architecture of DBN was obtained by training and stacking four layers with each layer holding 100 restricted Boltzmann machine (RBM). The architecture of the SDAE model incorporates three layers SDAE with each autoencoder stacked on the top of each other and each autoencoder having 2000, 1000, and 400 hidden neurons with corruption level of 0.5. The highest accuracy of 0.8119 was obtained in using DBN.

Nahid et al. [21] first used unsupervised clustering and further used the deep neural network models guided by the clustered information to classify the breast cancer images [27] into benign and malignant classes.

Wang et al. [35] proposed a deep learning-based system to auto-detect metastatic cancer from whole slide images of sentinel lymph nodes in the Camelyon challenge 2016. The authors compared GoogLeNet, AlexNet, VGG16, and FaceNet after pre-processing of excluding white background space. The highest performance was obtained in using GoogLeNet with 40 times magnification

Steiner et al. [30] conducted a study utilizing results from deep learning algorithms for the detection of breast cancer metastasis in lymph nodes. The study involved reviewing 70 slides by six pathologists in two modes assisted, and unassisted wherein the deep learning mode was used to outline interesting regions in assisted mode. The study found that algorithm-assisted pathologists demonstrated higher accuracy than either the algorithm or the pathologist alone. Pang et al. [22] proposed multiple magnification feature embedding (MMFE) as image tile prediction encoder and slice feature extractor. The method considered inputs image tiles in three resolution 256, 1024, and 4096 and scales to 256. The authors reported 78.1% accuracy in case of MMFE (tile results) and 84.6% accuracy in case of MMFE (features).

6. Conclusion

We found that some of the techniques did not improve the performances of the model. These techniques include progressive learning; focal loss [16]; average, geometric, and power weights for ensembles; training models with the center crop of 32 px instead of resized images.

Also, some of the models show significant improvement in the semi-supervised learning approach. With this approach, without the common k-fold method, the area under the curve (AUC) of a best single model could be increased after ten fine-tuning training runs from 0.971 to 0.9794 (evaluated on 100% of test data) outperforming the benchmark solution introduced in [33].

In general, pseudo labeling technique allows the training set to be enlarged without knowing the correct labels, allowing the model to achieve better generalization where entropy regularization [7] is a way to benefit from unlabeled data within the maximum a posteriori estimate. This opens up new possibilities for practical use of the model, the basic idea of which is that the single model could be continuously improved in the backend with unlabeled patches derived from new WSIs, which are uploaded to the frontend by the pathologist.

Acknowledgements

We would like to thank Frank Ihlenburg for his valuable comments and acknowledge Kaggle for availing the dataset for this work.

References

[1] Emile Aarts and Jan Korst. Simulated annealing and boltzmann machines. 1988. 2

[2] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. 1, 5

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009. 2

[4] H Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and P Muller. Deep neural network ensembles for time series classification. *arXiv preprint arXiv:1903.06602*, 2019. 5

[5] Jeffrey Alan Golden. Deep learning algorithms for detection of lymph node metastases from breast cancer: helping artificial intelligence be seen. *Jama*, 318(22):2184–2186, 2017. 1

[6] Lena Gorelick, Olga Veksler, Mena Gaed, José A Gómez, Madeleine Moussa, Glenn Bauman, Aaron Fenster, and Aaron D Ward. Prostate histopathology: Learning tissue component histograms for cancer detection and classification. *IEEE transactions on medical imaging*, 32(10):1804–1818, 2013. 5

[7] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 3, 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2

[12] Cheng Ju, Aurélien Bibaut, and Mark van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818, 2018. 5

[13] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 3

[14] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. *arXiv preprint arXiv:1801.05134*, 2018. 2

[15] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 2

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[17] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. 1

[18] Jiayi Liu, Samarth Tripathi, Unmesh Kurup, and Mohak Shah. Make (nearly) every neural network better: Generating neural network ensembles by weight parameter resampling. *arXiv preprint arXiv:1807.00847*, 2018. 5

[19] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q. Nelson, Gregory S. Corrado, Jason D. Hipp, Lily Peng, and Martin C. Stumpe. Detecting cancer metastases on gigapixel pathology images. *CoRR*, abs/1703.02442, 2017. 1, 4

[20] Vladimir Macko, Charles Weill, Hanna Mazzawi, and Javier Gonzalvo. Improving neural architecture search image classifiers via ensemble learning. *arXiv preprint arXiv:1903.06236*, 2019. 5

[21] Abdullah-Al Nahid, Mohamad Ali Mehrabi, and Yinan Kong. Histopathological breast cancer image classification by deep neural network techniques guided by local clustering. *BioMed Research International*, 2018. 6

[22] H. Pang, W. Lin, C. Wang, and C. Zhao. Using transfer learning to detect breast cancer without network training. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 381–385, Nov 2018. 6

[23] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013. 2

[24] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017. 2

[25] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018. 3

[26] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. 2018. 2

[27] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, 63(7):1455–1462, 2016. 6

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2

[29] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015. 2

[30] David F. Steiner, Robert MacDonald, Yun Liu, Peter Truszkowski, Jason D. Hipp, Christopher Gammage, Florence Thng, Lily Peng, and Martin C. Stumpe. Impact of deep learning assistance on the histopathologic review of lymph nodes for metastatic breast cancer, 2018. 6

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[32] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44, 2019. 1

[33] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018. 1, 4, 5, 6

[34] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016. 4

[35] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H. Beck. Deep Learning for Identifying Metastatic Breast Cancer. *arXiv e-prints*, page arXiv:1606.05718, Jun 2016. 6

[36] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In *International MICCAI Brainlesion Workshop*, pages 61–72. Springer, 2018. 4

[37] Wei Qian Wenqing Sun, Bin Zheng. Computer aided lung cancer diagnosis with deep learning algorithms, 2016. 5

[38] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. 2