Breast Cancer Metastases Identification using Deep Learning

Dong Liang, Li Ai and Shuang Zhou

ABSTRACT

Breast cancer patient if diagnosed in early stage has high 5-year survival rate. However, the tumor grading and cancer staging procedure which involves manually determined by pathologies under microscopes are time-consuming and prone to mistakes. This project proposed a two-staged identification model using deep learning to identify isolated tumor region in whole slide images of H&E stained lymph nodes slides and a targeted dynamic algorithm to strength as well as maintain model's accuracy

I. BACKGROUND

Breast cancer is the second most common cause of death from cancer in women in the United States. However, if the patient is diagnosed in early stage, the survival rate is very high. According to National Cancer Institute, the 5-year relative survival rate for localized breast cancer is 99%, whereas when cancer cells spreads to the nearby lymph nodes, the 5-year survival rate is 85%. If the cancer cells enter the distant parts of the body, the survival rate is only 27%. The detection and classification of cancer cells in nearly lymph nodes has significant meaning for breast cancer prognosis.

Surgically removed lymph nodes must go through a complex and long preparation and staining procedure in order to be examined under the microscope by pathologists. The size of the cancer cells and the number of cancer cells (if existed) are the determinants for what stage the cancer is and how far the cancer has progressed in the body. The most common and useful cancer staging system for invasive breast cancer is created by The American Joint Committee on Cancer. The system is called TNM staging system. T refers to tumor size, N refers to nearby lymph nodes status, M refers to metastasis in the distant lymph nodes.

Any work involved manually inspecting samples through microscope is time-consuming and prone to mistakes [1], as it is very difficult to detect small metastases. Spurred by recent development in deep learning and whole-slide scanner technology, Camelyon 16 [2] public challenge was organized in year 2016 by Radboud University Medical Center (Nijmegen, the Netherlands) and the University Medical Center Utrecht (Utrecht, the Netherlands) to evaluate new and existing algorithms for automated detection of metastases in stained whole-slide images of lymph node sections, and subsequently Camelyon 17 [3] was created in year 2017 by Diagnostic Image Analysis Group (DIAG) and Department of Pathology of the Radboud University Medical Center in Nijmegen, The Netherlands to further classify metastasis into three categories (Table 1). The best performing algorithm in Camelyon 16 performed equally well as a pathologist under no time constrain in terms of detection accuracy. The top-5 team in Camelyon 16 achieved AUC 0.9935 to 0.9234 for whole-slide-image classification and a composite score of 0.8074 to 0.6933 for tumor localization.

Table 1. Metastases categories

Category	**Size**
Macro-metastasis	Larger than 2 mm
Micro-metastasis	Larger than 0.2 mm and/or containing more than 200 cells, but not larger than 2 mm
Isolated tumor cells	Single tumor cells or a cluster of tumor cells not larger than 0.2 mm or less than 200 cells

II. RELATED RESEARCH

A typical WSI images under 40x resolution in Camelyon challenges has 200,000 x 10,000 pixels which calls for dimension reduction methods. Most of the current deep learning methods involve first filtering out tissue region as most area in WSI are blank, then training existing networks architectures on patch images extracted from tissue region. Wang et al. [4] trained a GoogLeNet based on randomly extracted 256 x 256 pixels patches to generate a prediction heatmap for each WSI, then apply random forest classifier to localize metastasis region and classify whole slide to having metastasis or negative. They also found out that 40x resolution performed better than other lower resolutions. This method has been widely adopted in Camelyon 17 participants. The top 12 performing team in Camelyon 17 used patch sizes ranging from 128 x 128 to 960 x

960[3]. To further optimize this method, Li et al. [5] proposed a selection model with clustering algorithm and CNN to select more discriminative patches.

One potential unsettling effect of this approach is that the resolution of heatmaps could potentially be much lower than original WSI. The cancer cells are microscopically small. Losing resolution could contribute to the low ITC detection accuracy in Camelyon 17. Guo et al. [6] proposed a model called v3_DCNN to combine patch extraction and pixel classification to perform a fast and refined metastasis region segmentation. They first filtered out tissue region, then 1280 x 1280 patches were extracted and applied to a slimmed-down Inception V3 to identify possible metastasis region which subsequently been fed through a dense deep convolutional neural network (DCNN) proposed by Chen et al [7]. Chen et al. devised an atrous convolution operation in place of traditional convolution operation to partly preserve the resolution of original image when going through layers of CNN.

III. DATA

The first-stage model uses the PatchCamelyon [8] benchmark data (https://github.com/basveeling/pcam) that consists of 327,680 color images (96x96) extracted from Camelyon 16 Challenge. Each image is annotated with a binary label indicating if the center 32x32 region contains at least one pixel of tumor issue or not. The ratio of the positive to negative patch is chosen to be close to 50/50 to keep the balance of the data.

Camelyon 17 images were used for second-stage model. The top four teams in Camelyon 17 achieved high accuracy (true positive) in identifying macro-metastasis and negative (no metastasis) slides but low accuracy rate in identifying ITC and confused negative as ITC with an alarming rate [3]. The recalls for ITC are only from 0% to 34.3%, and a large percentage (51.4% to 91.3%) of the ITC was mis-classified as negative. Therefore, due to the constraints of time and resources, we chose to only use the images that are classified as ITC and negative. Out of the 36 ITC WSI and 558 negative WSI, we only had time to compress 20 for ITC and 20 for negative for this course project.

IV. Methodology Part I: Proposal of a novel dynamic training algorithm

One of the major challenges of building an image classifier we encountered in this project is the very high resolution of medical whole slide images (WSI) leading to time-consuming and compute-intensive model optimization. We have to adopt a two-stage modeling strategy to first train a binary image classifier for cancerous cell patches, and then to build a secondary neural network to classify the cancer stage of the WSIs, however, at the cost of reducing image resolution. This means the training quality of the first image classifier could have a crucial effect on the optimization of subsequent models. In order to obtain better classification performance, in the first part of our study, we made a preliminary attempt to develop a more efficient training algorithm for general image classification purposes. The core idea is to train a model capable of identifying the latent features of error-prone images, and then compare the new training samples with these latent features to calculate the probability of these samples being misclassified by the image classifier. The composition of new training samples will accordingly be updated to include more error-prone samples for the new training.

Targeted dynamic training

If we view the training of an image classifier as a dynamic process, the training quality would depend to a large extent on the composition of the training samples. A good training process should take into account both consolidation and improvement. Especially in the final stage of training, when the model has learned most features and encountered a bottleneck in the accuracy rate, targeted dynamic training can strengthen the model by exposing it to more new error-prone samples. We here propose the prototype of a dynamic, self-balanced training algorithm as follows:

TARGETED DYNAMIC TRAINING

1. Randomly sample 1 % of test dataset → samplar; the remaining 99 % of test dataset → new test dataset.
2. Partition the samplar into two cohorts of misclassified and correctly classified samples based on the prediction of the image classifier.
3. Calculate the per-sample probability of being misclassified for the valid dataset (named ‘error-prone’).
4. Adjust the composition of the new training samples as follows: α % random training samples + $(100 - \alpha)$ % error-prone valid samples ($0 \leq \alpha \leq 100$), where $1 - \alpha$ is proportional to test accuracy of the image classifier (or simply set $\alpha = 0$)
5. Retrain the image classifier with the new training samples. Calculate the new test accuracy.

6. Repeat step 1- 5 until the test accuracy converges

Train a simple convolutional image classifier
We trained a simple VGG16 image classifier that achieved a test accuracy of 78.5%. Due to the page-limit we will not elaborate on it here, but the neural network training details (e.g. epoch, learning rate, network architecture, *etc.*) can be obtained in the source codes and PPT submitted together.

Identification of error-prone latent features
The key point of this proposed algorithm is to identify two sets of latent embeddings corresponding respectively to cohorts of correctly- and mis- classified samples and distinguish them as much as possible in the latent space. To achieve this goal, we implemented two distinct versions of Autoencoder and Siamese (not shown due to page limit) neural networks.

Autoencoder generated latent features: We trained a relatively simple Autoencoder with an architecture of three fully connected layers. The complete model architecture and training details are provided in the source code & the PowerPoint submitted separately. After 200 training epochs, we found that our Autoencoder successfully projected the high-resolution image features into a 10-dimensional latent space (**Fig. 1**). As shown in the figure below, the latent features of the error-prone samples have relatively low activation level in general, but sporadic activations occur in the 9th and the10th dimensions. In contrast, correctly classified samples present a much stronger and globally active patterns (**Fig. 1**). Due to time constraints, we did not further investigate which of the activated latent features correspond to the certain cancerous cell characteristics, but this would be a very interesting direction for future research.

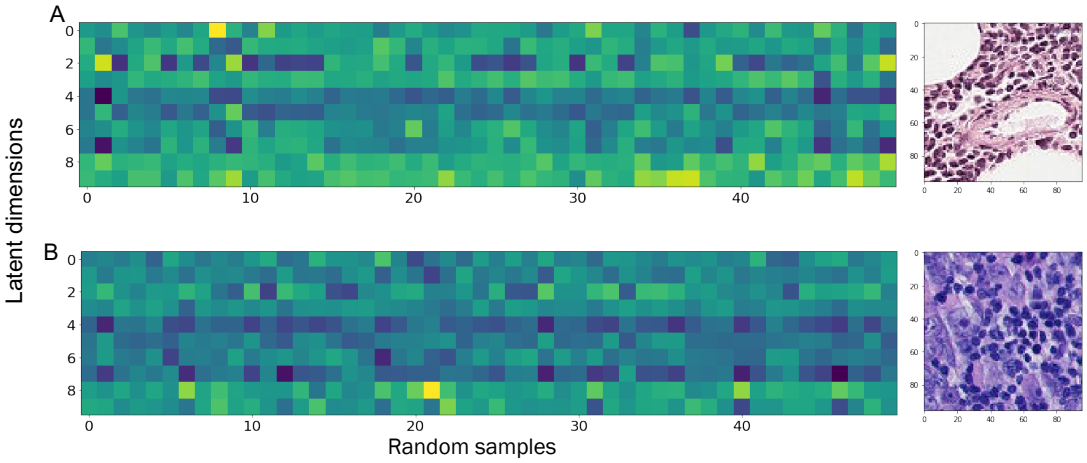

Figure 1. Autoencoder-generated latent features for correctly- **(A)** and mis- **(B)** classified images with sample images shown side by side on the right

Per-sample misclassification probability
In order to verify the validity of these Autoencoder-generated latent features, we further compared the latent embedding of new training samples with these two sets of latent features and calculated the probabilities of these samples being misclassified by the image classifier based on the cosine distances (**Fig. 2**). The per-sample misclassification probability was calculated as follows:

$$\text{Per-sample Probability} = \frac{1/\min(\text{Cosine\_distance1}(x))}{1/\min(\text{Cosine\_distance1}(x)) + 1/\min(\text{Cosine\_distance2}(x))}$$

where Cosine_distance1[or 2] (x) computes the cosine distances between latent embedding of testing sample x with all the mis-classification [or correct classification] latent features.

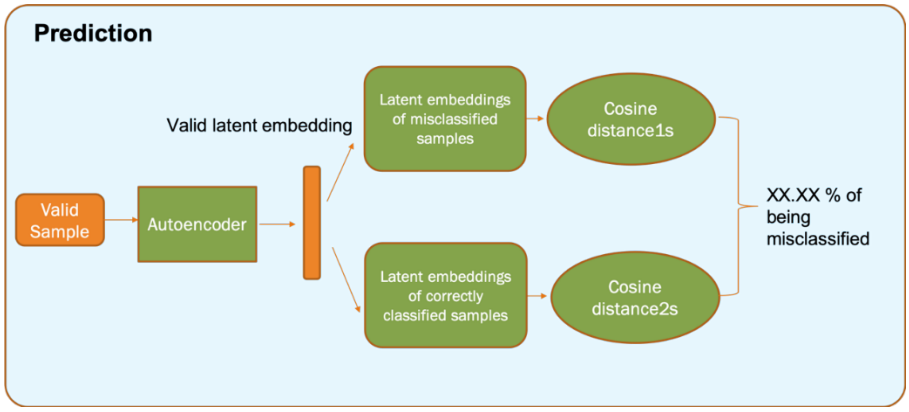

Figure 2. Calculation of per-sample misclassification probability based on cosine distance between latent embeddings.

Based on distribution of the per-sample misclassification possibilities (**Fig. 3A**), we divided the new training samples into three groups: two groups of samples with low ('Low prob') or high ('High prob') predicted misclassification rate, and one group of random samples ('Random'). These three groups of samples were predicted using the pretrained VGG16 image classifier (78.5% test accuracy), and the ground-truth labels were used to calculate the true misclassification rate. As shown in the figure below (**Fig. 3B**): the misclassification rates of 'Low Prob', 'Random', and 'High prob' groups are 0%, 19.1%, and 50%, respectively, showing that the misclassification rate of the 'High prob' group are noticeably higher than those of the other two groups.

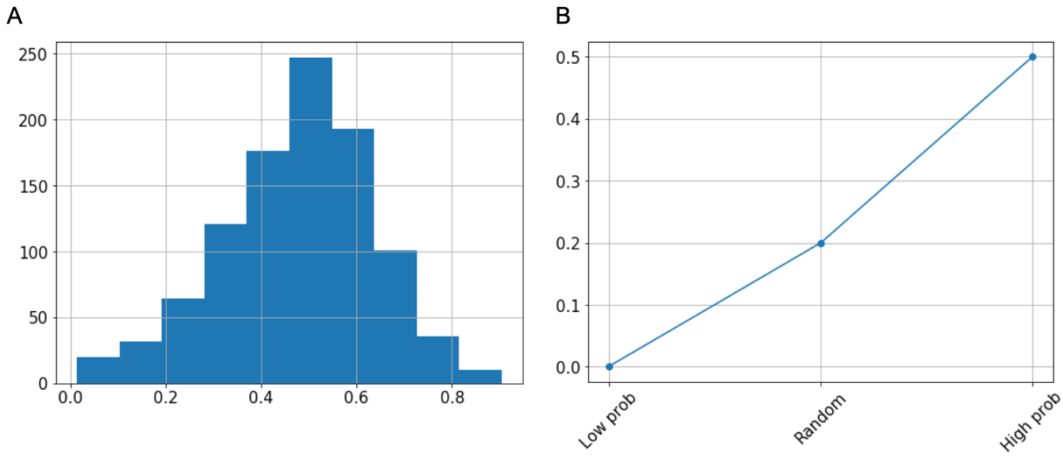

Figure 3. (A) Distribution of predicted misclassification probabilities. **(B)** True misclassification rate of the three groups of training samples

Algorithm validity:

We tried to test the validity of our algorithm using the complete set of cancerous cell patches (including more than 300,000 patches) on our local computer and/or Google Cloud Console platform but failed to get the final results as the training was always automatically interrupted due to memory limitations causing dead kernel issues. However, the source codes that can be used for testing is obtainable in the jupyter notebook we submitted together.

Advantage and limitation:

It can be expected that the Autoencoder version would work well with images that can easily form self-distinguishing clusters in the latent space (e.g. MNIST handwritten digit dataset). However, for images with complex features, different sets of latent features may overlap each other in the latent space, which will directly cause the same image, when situated in the overlapping area, to be classified into both categories with high cosine similarity. In practice, this will show as the misclassification probability being around 50%. In this case, the threshold of the possibility needs to be adjusted accordingly. For example, a threshold greater than 70% could be set for selecting images of interest. Additionally, the

Siamese network directly learns a similarity function by taking pairs of misclassified or correctly classified image samples, so it often does not require a lot of instances to achieve better prediction performance.

V. Methodology Part II: Two-stage image classification model

Modeling Strategy

Because the high-definition WSI's are too large to train using semantic segmentation algorithms such as U-net and DeepLab, we have decided to use a two-step approach: the first step is to build a binary classification model for classifying 96x96 image patches extracted from the WSI; the second step is to use the first-step model to score each 96x96 patch from a WSI, create a 2000x1000 probability map and then build another neural network to classify the cancer stage of the WSI. Essentially, the first model is used to 'compress' the original WSI into a smaller 'image' with reasonable size.

As can be expected, the overall performance of the two-stage modelling depends on the performance of the models in both stages. In the first stage, the first model scans through all the 96x96 patches (there are about 2 million of them in one WSI) to predict the probability of each patch being a cancer patch or not. Ideally, we would expect the first model to have an accuracy as high as possible. In reality, because of the highly heterogeneous nature of the WSI, it would be very difficult to use a single threshold (e.g. a naive 0.5) to separate cancerous and non-cancerous patch by the 1st-step model. For example, a non-cancerous area could be a benign breast cell, a nerve cell or glass slide. Therefore, we would like the first-stage model to have a strong separating power (e.g. high AUC) and use the second-stage model to 'learn' how to interpret the result from the first-stage model.

In the second stage of the model, the 2000x1000 image is first filtered using several different thresholds: any elements larger than the threshold are passed through as they are, while any elements smaller become zero. We chose this type of filtering because we would like the gradient can be passed back using back-propagation. If only binary values (0 and 1) are passed through the filter, no gradient can be passed back. In addition, because of the nature of the result from the first model, high threshold values are used, e.g. 0.9 and 0.99.

After filtration, the filtered images are trained using either fully-connected multilayer perceptrons or convolutional neural networks in parallel. The final layers of the neural networks for all the filtered images are then concatenated together as a weighted-mean, with the weight for each final layer being a variable the network needs to learn. Finally, the outputs are converted into a single scalar output for sigmoid transformation through fully-connected layers. The structures of the two proposed networks are illustrated in **Fig. 4**.

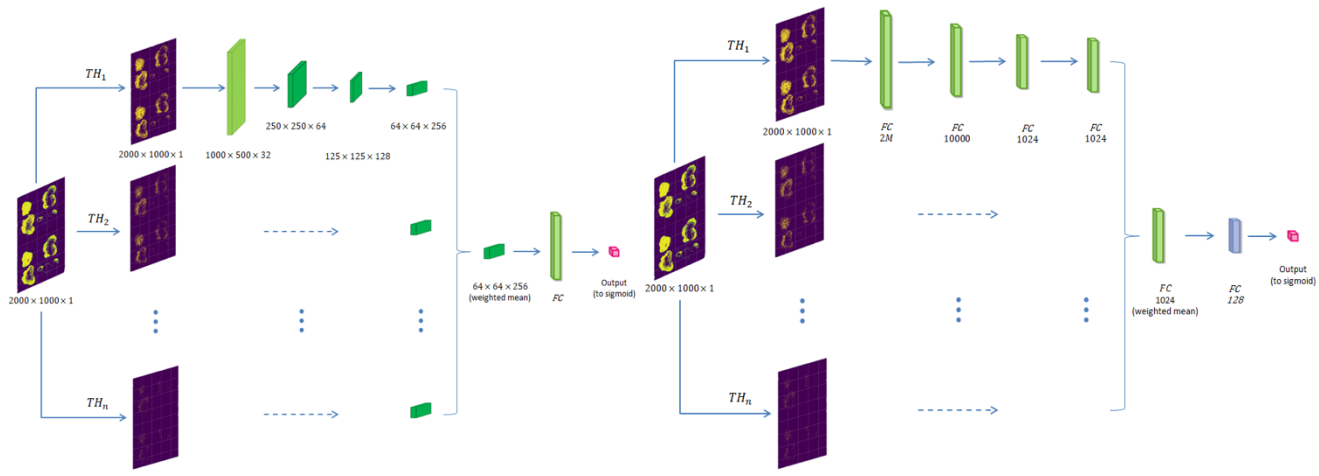

Figure 4. Proposed Architecture of (a) CNN and (b) Fully-Connected MLP for 2nd-Stage Model

Results

1st-Stage Model

Three commonly used CNN were trained from scratch for the train data: small VGGNet, ResNet50 and InceptionV3. We have performed a few trial-and-error training runs for a couple of epochs for these models and finally selected the following hyperparameters for training with the consideration of resources and efficiency (**Table 2**)

Table 2. Hyperparameters for Training the 1st-Stage Model

Learning Rate	Batch Size	Epoch
2.5 x10-5	128	30

The training history of the three models are shown in **Fig. 5**. As for VGGNet, the accuracy of valid data almost leveled off after 10 epochs, while the loss was still decreasing at 30 epochs. As for ResNet50, both accuracy and loss were still improving at 20 epochs. Compared to the train data, validation data showed a very uneven and zigzag pattern. One possible explanation is that, because of the heterogeneous nature of the images, the train and valid data was not separated with perfect stratification. In fact, there is much more negative WSIs as well as negative image regions/patches compared to the ITCs or the cancerous regions, and thus it would be challenging to create a dataset with 50/50 ratio.

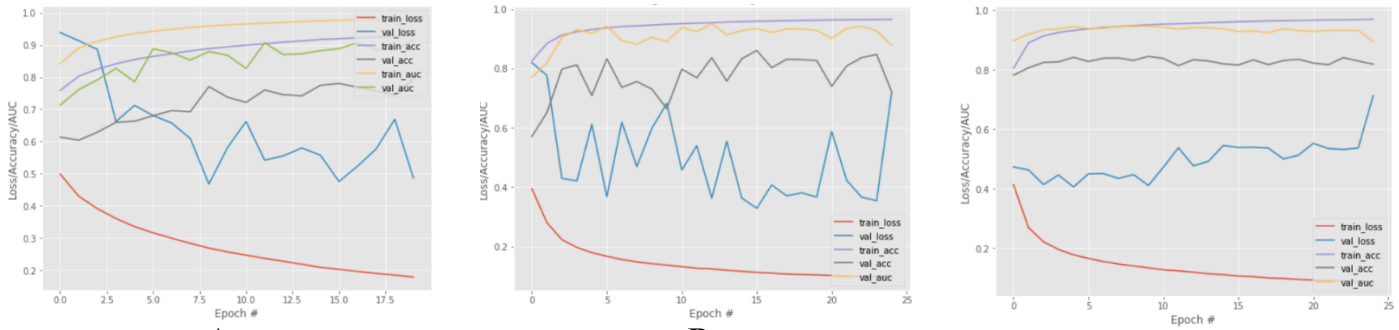

Figure 5. Loss, accuracy and AUC during training for: **A** small VGGNet; **B** ResNet50; and **C** InceptionV3.

Among the three model, InceptionV3 and ResNet50 have similar accuracy and AUC (**Table 3**). Compared to ResNet50, InceptionV3 has higher accuracy and AUC for test data and has a more monotonic training history. Therefore, InceptionV3 model is chosen as the benchmark model for the autoencoder-assisted selective training.

Table 3. Evaluation metrics from the benchmark models

CNN Architecture	Loss			Accuracy			AUC		
	Train	*Valid*	*Test*	*Train*	*Valid*	*Test*	*Train*	*Valid*	*Test*
small VGGNet	0.181	0.492	0.938	0.929	0.826	0.724	/	/	/
ResNet50	0.179	0.487	0.573	0.930	0.793	0.771	0.982	0.915	0.871
InceptionV3	0.138	0.411	0.510	0.951	0.845	0.807	/	0.946	0.934

2nd-Stage Model

Because of the limited number (20) of samples we have, our step-2 model only serves as a proof of concept. Because we did not have enough resources to run the CNN architecture, we only present the results for fully-connected MLP (**Table 4**).

Table 4. Hyperparameters for Training the 2nd-Stage Model: fully-connected MLP

No of hidden layers	No of neurons	Init method	Optimization	Dropout rate	Learning rate	Batch size	Epoch
2	1000	xavier	adam	0.5	0.0001	1	1

We have tried several different combinations of hyperparameters and found that the initialization method, learning rate and dropout rate are all important for the performance. As shown in **Fig. 6A** and **B**, the best loss with he-normal initialization is lower than that obtained from xavier-normal, as the number of input and output neurons in our model architecture can be very different in some layers. The impact of learning rate can be seen in **Fig. 6B** and **C**: when the learning rate increases from $1x10^{-7}$ to $1x10^{-6}$, the loss reaches minimum much faster, although it ends up being higher. We also learned from the experiment that a regularized model with higher dropout rate appears to have better performance, as demonstrated in **Fig. 6B** and **D.**

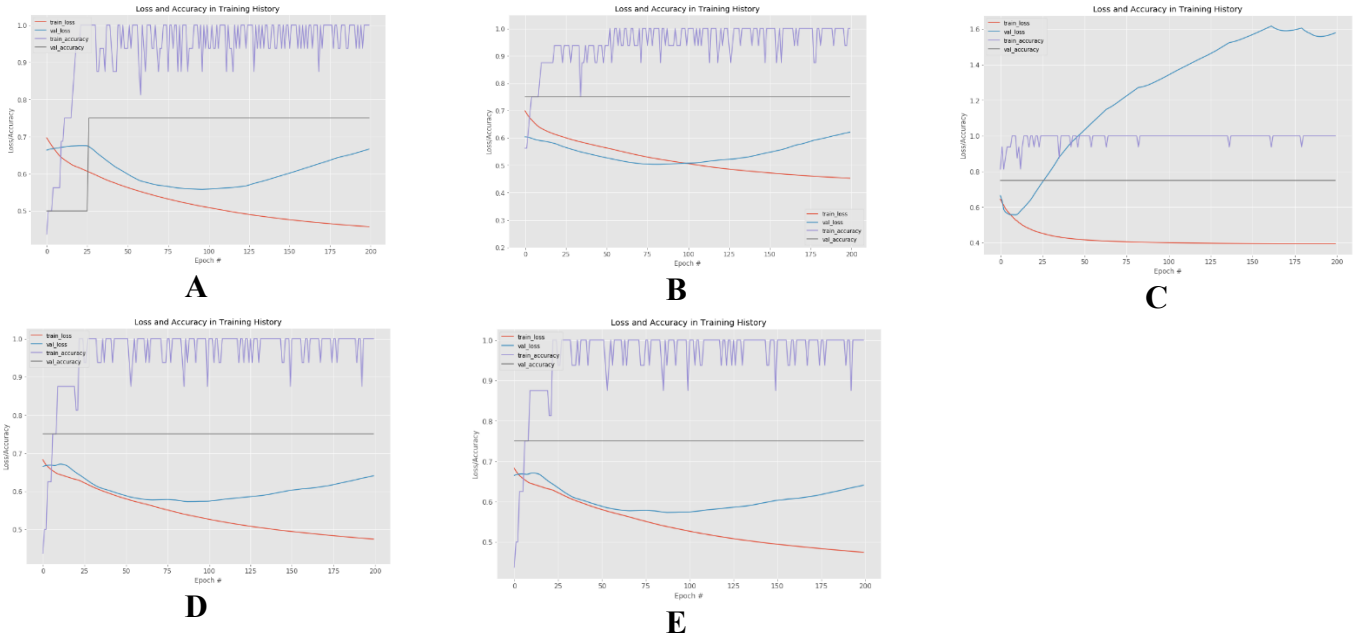

Figure 6. Training History of Fully-Connected MLP Model in Step 2: **A** init method = xavier normal, learning rate = $1x10^{-7}$ and dropout rate = 0.5; **B** init method = he normal and learning rate = $1x10^{-7}$ and dropout rate = 0.5; **C** init method = he normal and learning rate = $1x10^{-6}$ and dropout rate = 0.5; **D** init method = he normal and learning rate = $1x10^{-7}$ and dropout rate = 0.01.

We have only used a small number of neurons (50) and layers (3) in our experiment, as permitted by the resources. We would expect the performance of the model to be higher when more complicated models (more neurons, more layers, more complicated architecture, etc.) and the full dataset (200 WSI) are used in training.

VI. CONCLUDING REMARKS

In this project, we developed an innovative dynamic training algorithm, called targeted dynamic training to expose model to new error-prone data in order to gradually improve its accuracy among mis-identified data. At each exposure, positively identified data also being fed into the model in order to maintain model's accuracy with positive data. We also developed a 2-stage identification model employed smallest patch size among published researches. Due to time and resource constrains, methodology in this project is limited as proof of concept.

It has been a discovery journey for us, technology wise and reality wise. We choose this project because it has a real-life application. It has potential to save people's lives by expanding the limitation and horizon of what humans can do in terms of efficiency and accuracy. It is the manifestation of what's represented in the saying of Technology changes life. However, this project also showcased how heavily AI application like this relies on hardware. Whole slide scanners, high performance computer clusters with GPUs, rewiring infrastructure required intensive capital investment. There are probably only a handful of pathology labs have the financial means to adopt this application. Giant IT companies, Google and Amazon both provides remotely accessible GPUs as a solution, but not for project like this involving deep learning on large images. From our experience Google Cloud is unstable and failed to run any of the algorithm in our identification model.

History repeats itself. There were times in early days of personal computers history that people had to buy a terminal and blocks of computer time in a remote system. We imagine that there will be a technology breakthrough which will lead to make AI more affordable and accessible to the general public. Only until that time, the power of AI will truly be harvested by human to make our lives better and change our lives in an unforeseeable manner.

VII. REFERENCE

[1] L. He et al., “Histology image analysis for carcinoma detection and grading”, Comput Methods Programs Biomed, vol. 107, no. 3, pp. 538-556, Sep. 2013

[2] B.E.Bejnordi et al., “Diagnostic assessment of deep learning alorithms for detection of lymph node metastases in women with breast cancer”, *JAMA*, vol. 318, no. 22, pp. 2199-2210, Dec. 2017

[3] P. Bandi et al., “From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient level: The CAMELYON 17 Challenge”, *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 550-560, Feb. 2019

[4] D. Wang et al., “Deep learning for identifying metastatic breast cancer”, *arXiv prepring arXiv: 1606.05718*, 2016

[5] Y. Li et al., “ Classification of breast cancer histology images using multi-size and discriminative patches based on deep learning”, IEEE Access, 7, doi: 10.1109/ACCESS.2019.2898044, Feb 2019

[6] Z. Guo et al., “A fast and refined cancer regions segmentation framework in whole-slide breast pathological images”, *Sci Rep* 9, 882, doi: 10.1038/s41598-018-37492-9, 2019

[7] L. -C., Chen et al., “ DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs”, arXiv: 1606.00915

[8] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling. "*Rotation Equivariant CNNs for Digital Pathology*". arXiv:1806.03962