

PDBL: Improving Histopathological Tissue Classification With Plug-and-Play Pyramidal Deep-Broad Learning

Jiatai Lin, Guoqiang Han^{ID}, Xipeng Pan, Zaiyi Liu, Hao Chen^{ID}, Danyi Li, Xiping Jia^{ID}, Zhenwei Shi^{ID}, Zhizhen Wang, Yanfen Cui, Haiming Li, Changhong Liang, Li Liang, Ying Wang, and Chu Han^{ID}, Member, IEEE

Abstract— Histopathological tissue classification is a simpler way to achieve semantic segmentation for the whole slide images, which can alleviate the requirement of pixel-level dense annotations. Existing works mostly leverage the popular CNN classification backbones in computer vision to achieve histopathological tissue classification. In this paper, we propose a super lightweight plug-and-play

Manuscript received January 17, 2022; revised March 15, 2022; accepted March 20, 2022. This work was supported in part by the Key Research and Development Program of Guangdong Province, China, under Grant 2021B0101420006; in part by the National Key Research and Development Program of China under Grant 2021YFF1201003 and Grant 2021YFF1201004; in part by the National Science Fund for Distinguished Young Scholars under Grant 81925023; in part by the National Natural Science Foundation of China under Grant 62102103, Grant 82071892, Grant 82001789, Grant 81901704, Grant 82102034, and Grant 62002082; in part by the High-level Hospital Construction Project under Grant DFJH201805 and Grant DFJHBF202105; and in part by the Guangzhou Research and Development Plan in Key Areas under Grant 20200704001. (*Jiatai Lin, Guoqiang Han, and Xipeng Pan contributed equally to this work.*) (*Corresponding authors: Li Liang; Ying Wang; Chu Han.*)

Jiatai Lin and Guoqiang Han are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: linjiatai_cs@163.com; csgqhan@scut.edu.cn).

Xipeng Pan, Zaiyi Liu, Zhenwei Shi, Yanfen Cui, Haiming Li, Changhong Liang, and Chu Han are with the Department of Radiology and the Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, 510080 China (e-mail: pxp201@guet.edu.cn; zyliu@163.com; zhenwei_shi88@163.com; yanfen210@126.com; lihaiming0109@163.com; liangchanghong@gdph.org.cn; hanchu@gdph.org.cn).

Hao Chen is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (e-mail: jhc@cse.ust.hk).

Danyi Li and Li Liang are with the Guangdong Province Key Laboratory of Molecular Tumor Pathology, Department of Pathology, Nanfang Hospital and Basic Medical College, Southern Medical University, Guangzhou, Guangdong 510515, China (e-mail: lidanyi26@163.com; lli@smu.edu.cn).

Xiping Jia is with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China (e-mail: jiaxp@gpnu.edu.cn).

Zhizhen Wang is with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guangxi 541004, China (e-mail: wangzhizhen0012021@163.com).

Ying Wang is with the Department of Medical Ultrasonics, The First Affiliated Hospital of Guangzhou Medical University, Guangzhou, Guangdong 510120, China (e-mail: liuyiy527@163.com).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TMI.2022.3161787>, provided by the authors.

Digital Object Identifier 10.1109/TMI.2022.3161787

module, named Pyramidal Deep-Broad Learning (PDBL), for any well-trained classification backbone to improve the classification performance without a re-training burden. For each patch, we construct a multi-resolution image pyramid to obtain the pyramidal contextual information. For each level in the pyramid, we extract the multi-scale deep-broad features by our proposed Deep-Broad block (DB-block). We equip PDBL in three popular classification backbones, ShuffLeNetV2, EfficientNetb0, and ResNet50 to evaluate the effectiveness and efficiency of our proposed module on two datasets (Kather Multiclass Dataset and the LC25000 Dataset). Experimental results demonstrate the proposed PDBL can steadily improve the tissue-level classification performance for any CNN backbones, especially for the lightweight models when given a small amount of training samples (less than 10%). It greatly saves the computational resources and annotation efforts. The source code is available at: <https://github.com/linjiatai/PDBL>.

Index Terms— Pyramidal deep-broad learning, histopathological tissue classification, broad learning system.

I. INTRODUCTION

HISTOPATHOLOGICAL slides not only play a vital role in cancer diagnosis, but also deliver valuable tumor microenvironment information for cancer research [1], [2]. To analyze the whole slide images by computer algorithms is crucial for precision medicine on cancers, such as diagnosis prediction [3], [4], molecular status prediction [5], [6] and even the origins of the unknown primary cancers prediction [7]. Segmenting and recognizing various tissue types is the very first step of histopathological image analysis. Semantic segmentation [8] is the best way to define tissue types for every single pixel. However, due to the gigapixel resolution and the expertise requirement, obtaining pixel-level annotations is extremely difficult and time-consuming [9]. Therefore, patch-level classification now becomes an alternative solution [10], [11], which can greatly save the annotation efforts.

Convolutional neural network (CNN) has demonstrated outstanding performance in image classification problem [12], with a series of classification backbones, i.e., ResNet [13], ShuffLeNet [14], EfficientNet [15] and etc. They have been rapidly extended to medical image classification [16], [17], including histopathological image classification [18], [19].

Typically, Han *et al.* [20] presented a CNN-based multi-classification model for histopathological tissue classification of breast cancer, which achieves over 94% patch-level accuracy under four magnification factors. Tsai and Tao [21] tested five common classification backbones on colorectal tissue classification using 100,000 training image patches. All the backbones can achieve over 95% accuracy. The above researches proved that the current CNN classification backbones have already demonstrated strong feature representation ability and achieved promising results for histopathology tissue classification. In this paper, we reconsider how to make good use of the features extracted from the existing CNN backbones, to further improve the classification performance as well as to increase the model generalizability, adaptability, and robustness.

In clinical practice, pathologists read histopathological slides by switching the object lens to observe the slides under different magnifications. Therefore, considering multi-scale contextual information is critical for histopathological image analysis. Inspired by this observation, we propose a lightweight plug-and-play module for any CNN classification backbones, named Pyramidal Deep-Broad Learning (PDBL). For each input image, an image pyramid is constructed to extract the pyramidal contextual information. For each level in the pyramid, we propose a Deep-Broad block (DB-block) to fully discover the multi-scale deep-broad features extracted by the CNN backbones from low level to high level. Our proposed PDBL can be plugged on any classification backbone and effectively improve the classification performance with very few extra computational resources.

We test PDBL on three representative CNN backbones, ShuffLeNetV2, EfficientNetB0, and ResNet50 on Kather Multiclass Dataset [11] and Lung Colon Cancer Histopathological Image Dataset [22]. We conduct two main experiments, one is the effectiveness of PDBL with different proportions of the training samples, the other is the robustness of PDBL with only 1% training samples. Experimental results demonstrate that PDBL effectively improves the classification performance on both datasets. When very limited training samples are involved (1% only), PDBL can maintain a standout improvement compared with the baseline models without PDBL. It can significantly reduce the annotation efforts and computational resources. Experimental results also show that PDBL improves domain adaptation abilities for CNN models. The contributions of this paper are summarized as follows:

- We propose a lightweight plug-and-play module (PDBL), which can be easily applied on almost any common CNN-based classification backbone. It can generally improve all the three CNN backbones in the experiment for histopathological tissue classification with no re-training burden.
- We propose a Deep-Broad block to fully discover the multi-scale deep-broad features from low level to high level.
- The proposed PDBL demonstrates outstanding improvement of the performance for the lightweight models with very limited training samples (1% only).

- Models with PDBL can relieve the requirement of large-scale training data and be easily and efficiently adapted to a new domain with only a few training samples, which greatly saves the computational resources and annotation efforts.

II. RELATED WORKS

A. Histopathological Image Classification

Automated analysis of whole slide images (WSIs) plays a crucial role in computer-assisted tumor diagnosis [3], [23]. Due to the giga-pixel resolution, directly processing the entire WSI is not feasible. Moreover, obtaining pixel-level annotation is extremely difficult. Hence, histopathological tissue classification has been widely employed as an alternative solution for tissue semantic segmentation of WSIs [11].

With the development of CNN models, most of the histopathological image classification models [24], [25] are originated from the popular classification backbones from the natural image classification. However, histopathological image classification faces different challenges, such as extremely large image resolution, deficiency of labels and multi-scale information integration [26]. WSI-Net [27] model was proposed to add an additional classification branch to discard the normal tissue in order to save computational resources. Raczkowski *et al.* [28] proposed a pathologist-in-the-loop model to solve the insufficient labeling problem. Xue *et al.* [10] proposed to synthesis histopathological patch images using GAN to enhance the feature representation and to improve the classification performance. Many studies attempted to extract multi-scale features to better solve the classification problem of histopathological images with end-to-end deep learning models, such as the Deep-Hipo model [29] and multi-resolution model [30]. Hatami *et al.* [31] proposed a pluggable Deep Multi-Resolution Dictionary Learning (Deep MRDL) based on the Deep Texture Encoding Network (Deep TEN) [32] to utilize the multi-scale features from the CNN backbone to improve the performance of baselines.

In this paper, we use a broad learning strategy to fully discover the deep features extracted by deep learning models and leverage the multi-scale contextual information to improve the performance of the CNN-based models without excessive computational costs. Next, we will introduce some common deep learning architectures and broad learning system approaches.

B. Deep Learning Architectures

Deep learning models have already dominated the image classification problem [33]. They usually go through several stages to reduce the feature dimension and to extract higher-level semantic features, defined in Eq. 1.

$$\mathcal{F} = \text{stage}_1 \odot \text{stage}_2 \odot \dots \odot \text{stage}_j(X), \quad j = 1, 2, \dots, h. \quad (1)$$

where \mathcal{F} denotes the CNN model and X is the input image. Each stage is composed of a series of cascaded convolutional blocks, such as Res-block [13], Efficient-block [15],

Shuffle-block [34] and Inception-block [35]. These blocks were designed to prevent the gradient vanishing problem and to increase the capacity of CNN models by balancing the depth and width of the deep architecture. Some skip connections were also introduced to transmit the features between different convolutional layers to avoid information loss and enhance the feature representation.

The current CNN classification backbones have already had strong capacity and feature representation ability. So in this paper, we aim to discover the potential of the multi-scale deep features extracted from different stages, and further improve the performance of any well-trained CNN backbones for histopathological tissue classification.

C. Broad Learning System

With the breakthrough of the GPU architectures over the past decade, researchers have kept increasing the depth of CNN models and achieved outstanding performance in most of the computer vision and medical imaging tasks. However, a deeper network will also increase the computational time of the model training. Chen *et al.* [36] proposed an opposite direction of neural network by expanding the width instead of increasing the depth, called Broad Learning system (BL). BL tends to breadth-wise expand the feature space by multi-group feature mapping and uses a shallow fully-connection layer to calculate output, which greatly saves the computational resources compared with deep learning (DL).

In the past years, a series of BL approaches [37]–[40] have been proposed. The motivation of these approaches is to provide an architecture to breadth-wisely combine multiple groups of features by solving the following optimization problems:

$$W_{opt} = \arg \min_{W_{init}} \| AW_{init} - Y \|_2^2 + \gamma \| W_{init} \|_2^2 \quad (2)$$

where Y , A represent target matrix (vector) and combined feature matrix that concatenates with all groups of feature nodes and enhance nodes. W_{init} and W_{opt} are pre-update and post-update weights of output layer, which can be updated rapidly by pseudo-inverse method:

$$W_{opt} = A^+ Y = \lim_{\lambda \rightarrow 0} (A A^T + \lambda E)^{-1} A^T Y \quad (3)$$

where E and λ represent an identity matrix and a constant parameter. When $\lambda = 0$, the updating method is linear regression which requires A to be a non-singular matrix. The meaning of the above symbols can be found in Table I.

In short, DL has a stronger semantic feature extraction ability, while BL is faster and more lightweight. So in this paper, we want to gather the strengths of both DL and BL by using DL to extract multi-scale semantic features and using BL for inference. Such deep-broad design is effective and will not introduce an extra training burden. Considering histopathological tissue classification, we design a pyramidal structure for the deep-broad learning to consider the pyramidal contextual information of the histopathological images.

TABLE I
SYMBOL ANNOTATIONS

Symbol	Meaning
I	Input image
X	Sub-image of the image pyramid
\mathcal{X}	Image pyramid
f	Intermediate deep features
e	Channel-wise feature vector of f
z	Multi-scale deep-broad feature vector of X
b	Pyramidal deep-broad feature vector of \mathcal{X}
B	Feature matrix of the complete training data
C	Covariance matrix
U	Dimension reduction matrix
A	Feature matrix after PCA
W_{PDBL}	Weights of PDBL
n	Number of the training samples
c	Number of the categories
q	Dimension of z
p	Dimension of b
d	Target dimension in dimensionality reduction

III. PYRAMIDAL DEEP-BROAD LEARNING

Deep learning (DL) has the powerful feature extraction ability while Broad Learning (BL) is good at combining multiple groups of features for fast inference. Theoretically, associating DL with BL can effectively improve the performance of existing CNN-based models. In this section, we propose a novel Pyramidal Deep-Broad Learning (PDBL) with a Deep-Broad block (DB-block) for histopathological tissue classification. Fig. 1 demonstrates the overview of the proposed PDBL. We first construct an image pyramid for the input image. And then we extract the multi-scale deep-broad features by DB-block, shown in Fig. 2. Finally, a broad learning system is introduced for the inference. The annotations in this article are defined in Table I.

A. Image Pyramid Construction

Typically, pathologists observe pathological sections under different magnifications. Inspired by this, we want the CNN-based models to be able to consider the multi-scale contextual information too. Given an input WSI patch I , we construct an image pyramid by downscaling the input image for s times, defined as follows:

$$X_i = \xi(I, S_i), \quad i = 1, \dots, s \quad (4)$$

where ξ denotes the scale transformation of bilinear interpolation with the scaling factor S_i .

Now we have an image pyramid \mathcal{X} with s sub-images, including the input image $X_1 = I$.

$$\mathcal{X} = \{X_1, X_2, \dots, X_s\} \quad (5)$$

And then, each sub-image is passed into our proposed DB-block for feature extraction.

B. Multi-Scale Deep-Broad Feature Extraction via Deep-Broad Block

We propose a Deep-Broad block (DB-block) to extract features for the image pyramid \mathcal{X} , as shown in Fig. 2. The

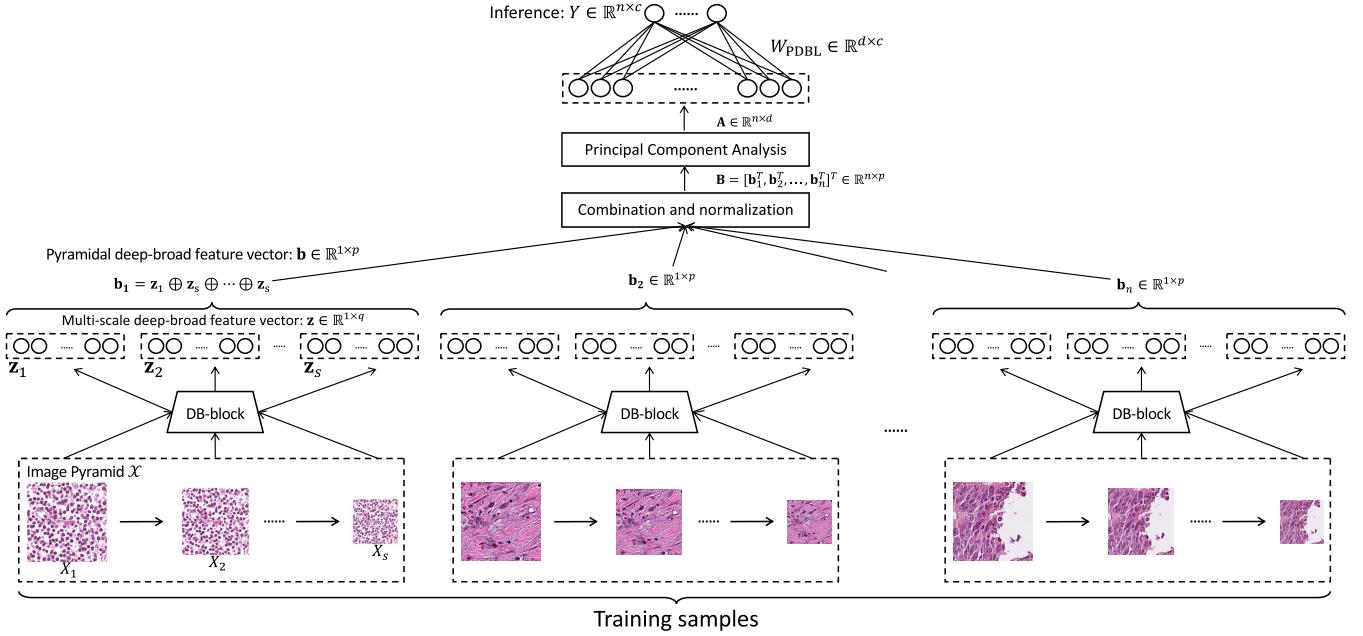


Fig. 1. Overview of the proposed Pyramidal Deep-Broad Learning (PDBL). In PDBL, we create an image pyramid for each image in the training set to obtain the pyramidal contextual information. For each image in the pyramid, we extract the multi-scale deep-broad features by a Deep-Broad block (DB-block). Finally, histopathological image categories can be inferred by the broad learning system. For simplification, we only show the notations of the left input image.

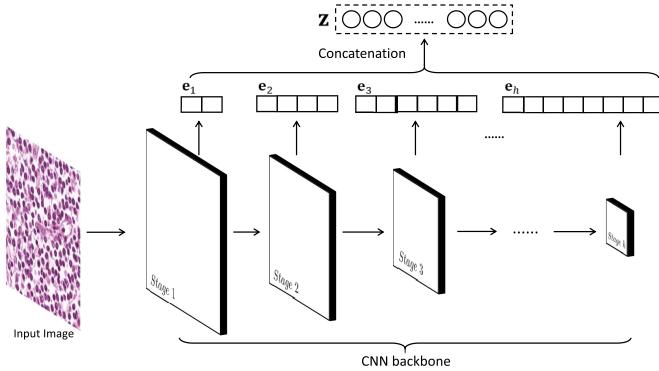


Fig. 2. Illustration of DB-block. Given any CNN backbone, intermediate feature maps of each stage are compressed into channel-wise features \mathbf{e} by the adaptive global average pooling. Then we can obtain the multi-scale deep-broad features \mathbf{z} by concatenating \mathbf{e} .

DB-block broadens the deep features from each stage of the CNN backbone and forms the multi-scale deep-broad features.

For the sub-image X_i , we first extract its intermediate deep features f from the last layer of each stage of the CNN backbone. The intermediate deep features f_k at the k -th stage are squeezed into a channel-wise feature vector by adaptive global average pooling in Eq. 6.

$$\mathbf{e}_k = \frac{1}{H_k \times W_k} \sum_{i=1}^{H_k} \sum_{j=1}^{W_k} f_k(i, j) \quad (6)$$

where H_k and W_k represent the height and width of the intermediate feature maps at the k -th stage.

The multi-scale deep-broad features \mathbf{z}_i of the sub-image X_i can be obtained by concatenating all the channel-wise feature vector \mathbf{e} , as follows:

$$\mathbf{z}_i = \mathbf{e}_1 \oplus \mathbf{e}_2 \oplus \dots \oplus \mathbf{e}_h \quad (7)$$

where $\mathbf{z}_i \in \mathbb{R}^{1 \times q}$, and q represents the dimensions of this feature group, which is the summation of channel numbers of all the stages. h denotes the number of stages of the CNN backbone. \oplus represents the concatenation operation.

Now for the image pyramid $\mathcal{X} = \{X_1, X_2, \dots, X_s\}$, we can obtain pyramidal deep-broad feature vector \mathbf{b} with pyramidal contextual information by concatenating all the multi-scale deep-broad features \mathbf{z} of the sub-images as follows:

$$\mathbf{b} = \mathbf{z}_1 \oplus \mathbf{z}_2 \oplus \dots \oplus \mathbf{z}_s \quad (8)$$

where $\mathbf{b} \in \mathbb{R}^{1 \times p}$ is the pyramidal deep-broad feature vector of the image pyramid \mathcal{X} . And p is the dimension of \mathbf{b} where $p = s \times q$.

In DB-block, we extract the deep learning features by the baseline model pre-trained with only training the CNN backbone once using the original training set. And we extract features of all the sub-images from the image pyramid by this CNN backbone.

C. Broad Learning Inference

With the pyramidal deep-broad feature vector \mathbf{b} of the image pyramid \mathcal{X} , we apply a broad learning system for inference. Let us denote the complete training samples as $\mathbb{X} = \{\mathcal{X}_i | i = 1, 2, \dots, n\}$, where n is the number of training samples. We can obtain a set of feature vectors $\{\mathbf{b}_i | i = 1, 2, \dots, n\}$. Then a broad feature matrix \mathbf{B} is constructed by combining all the feature vectors \mathbf{b}_i in Eq. 9.

$$\mathbf{B} = \delta([\mathbf{b}_1^T, \mathbf{b}_2^T, \dots, \mathbf{b}_n^T]^T), \quad \mathbf{B} \in \mathbb{R}^{n \times p} \quad (9)$$

where δ denotes the matrix normalization transformation.

In order to reduce the feature dimension and redundancy, principal component analysis (PCA) is employed for \mathbf{B} .

We first calculate the covariance matrix \mathbf{C} by Eq. 10:

$$\mathbf{C} = \frac{1}{n} \mathbf{B}^T \mathbf{B}, \quad \mathbf{C} \in \mathbb{R}^{p \times p}. \quad (10)$$

Furthermore, dimension reduction matrix is obtained by SVD algorithm:

$$[\mathbf{U}, \Sigma, \mathbf{V}^T] = SVD(\mathbf{C}). \quad (11)$$

where $\mathbf{U} \in \mathbb{R}^{p \times d}$, and d represents the target dimension in dimensionality reduction.

According to \mathbf{U} , the dimension of feature matrix \mathbf{B} can be reduced to the matrix \mathbf{A} by Eq. 12:

$$\mathbf{A} = \mathbf{B}\mathbf{U}, \quad \mathbf{A} \in \mathbb{R}^{n \times d}. \quad (12)$$

Finally, the probabilities Y of all the categories can be calculated by:

$$Y = \mathbf{A}W_{\text{PDBL}}, \quad W_{\text{PDBL}} \in \mathbb{R}^{d \times c} \quad (13)$$

where W_{PDBL} is the weights of PDBL, which can be calculated by the pseudo-inverse algorithm. c denotes the number of categories.

In the Training Phase: we use $\mathbf{A}_{\text{train}}$ to represent the feature matrix for the complete training set after PCA. According to the ground truth labels Y_{train} , the weights W_{PDBL} of PDBL can be calculated by the pseudo-inverse algorithm as follows:

$$W_{\text{PDBL}} = \mathbf{A}_{\text{train}}^+ Y_{\text{train}} \quad (14)$$

where $\mathbf{A}_{\text{train}}^+$ can be calculated by:

$$\mathbf{A}_{\text{train}}^+ = \lim_{\lambda \rightarrow 0} (\lambda E + \mathbf{A}_{\text{train}} \mathbf{A}_{\text{train}}^T)^{-1} \mathbf{A}_{\text{train}}^T \quad (15)$$

where λ and E represent a constant and unit matrix, respectively. Since the pseudo-inverse algorithm only updates the weights once, it greatly saves the computational resources.

In the Testing Phase: we can obtain the feature matrix \mathbf{A}_{test} by the same steps of the training phase. Then we can infer the probabilities Y_{test} of tissue categories by:

$$Y_{\text{test}} = \mathbf{A}_{\text{test}} W_{\text{PDBL}} \quad (16)$$

The final classification result is the tissue categories with the largest probabilistic value.

Since the proposed PDBL is a plug-and-play module. It can be applied to any CNN backbone and further improve the classification performance.

D. Implementation and Training Details

In our experiments, all the CNN backbones are implemented in PyTorch on a workstation with an NVIDIA RTX 2080Ti and a dual-core Intel i5 CPU. The backbones are trained with the cross-entropy loss and the SGD optimizer with a learning rate of $1e^{-3}$, weight decay of $1e^{-4}$ and momentum of 0.9. The batchsize is set to 20. Patches are resized into 224×224 and normalized in both training and test phases. All the CNN backbones in all the experiments in both datasets use the same hyperparameter setting.

For our proposed PDBL, we create image pyramid under three different resolution (112×112 , 160×160 and 224×224). The deep-broad features of the images of different scales are

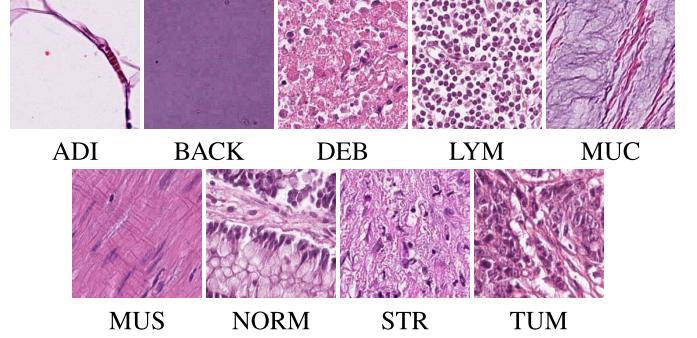


Fig. 3. Kather Multiclass Dataset includes Adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM).

all extracted from the network trained by the original image size (224×224). And the target dimension d of PCA is decided by the total number n of the training samples as follows:

$$b = \begin{cases} 0.9 * n, & n \leq 2000 \\ 2000, & n > 2000 \end{cases} \quad (17)$$

where n is the number of training samples.

IV. DATASETS

We evaluate our proposed PDBL in the following two datasets Kather Multiclass Dataset [11] and LC25000 Dataset [22].

A. Kather Multiclass Dataset

Kather Multiclass Dataset is a multi-class colorectal dataset composed of H&E stained histopathological tissue patches, which was published by J. N. Kather.¹ Kather Multiclass Dataset is composed of two subsets. Kather *et al.* [11] manually delineated tissue regions in 86 colorectal (CRC) tissue slides and they extracted 100,000 H&E histopathological tissue patches from these regions as Kather Multiclass Internal (KMI) subset at $20\times$ magnification. They also extracted an additional independent Kather Multiclass External (KME) subset including 7180 H&E stained histopathological patches. Histopathological images in Kather Multiclass Dataset are cropped to a square size of 224×224 pixels from the whole slide images (WSIs) at $20\times$ magnification.

As shown in Fig. 3, each histopathological image belongs to one category of tissues and there are 9 categories of tissues in Kather Multiclass Dataset, including adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), and colorectal adenocarcinoma epithelium (TUM).

B. LC25000 Dataset

To advance computer-aided automated analysis of lung and colon carcinomas, [22] released a lung and colon histopathological image dataset (LC25000 Dataset).² In LC25000

¹<https://zenodo.org/record/1214456>

²https://github.com/tampapath/lung_colon_image_set

TABLE II

QUANTITATIVE COMPARISON (ACC) WITH DIFFERENT PROPORTIONS OF TRAINING SAMPLES IN KATHER DATASET (100% TRAINING SAMPLES: 100,000 PATCHES) AND LC25000 DATASET (100% TRAINING SAMPLES: 15,000 PATCHES)

Models	Dataset	Accuracy							
		1%	5%	10%	25%	35%	50%	70%	100%
ShuffLeNetV2	Kather	0.17173	0.20362	0.36142	0.64359	0.65933	0.68468	0.71630	0.74805
		0.86880	0.92242	0.94095	0.94666	0.94568	0.94791	0.95139	0.95097
		0.17187	0.73607	0.91448	0.94805	0.96058	0.96114	0.95097	0.95195
		0.87869	0.93816	0.95696	0.96114	0.96365	0.96253	0.96198	0.96156
EfficientNetb0	LC25000	0.52000	0.81380	0.79300	0.83290	0.84150	0.84170	0.85090	0.85250
		0.93260	0.96730	0.97070	0.98670	0.98950	0.99300	0.99320	0.99400
		0.39280	0.62380	0.78980	0.96810	0.97310	0.98480	0.98890	0.99660
		0.94140	0.95970	0.96540	0.98990	0.99140	0.99370	0.99520	0.99720
ResNet50	Kather	0.70641	0.78078	0.81128	0.84248	0.84554	0.84847	0.85919	0.86309
		0.86685	0.90933	0.92214	0.92841	0.93064	0.93036	0.93287	0.93398
		0.91267	0.93788	0.94109	0.94011	0.94415	0.93816	0.93524	0.94471
		0.92256	0.94930	0.94847	0.94930	0.95432	0.95557	0.95641	0.96086
ResNet50	LC25000	0.83430	0.88270	0.90680	0.92270	0.92820	0.93290	0.93860	0.94290
		0.94170	0.97290	0.98200	0.99000	0.99360	0.99450	0.99530	0.99500
		0.87220	0.96320	0.98100	0.98960	0.99230	0.99630	0.99780	0.99870
		0.95300	0.97990	0.98750	0.99460	0.99610	0.99710	0.99860	0.99940

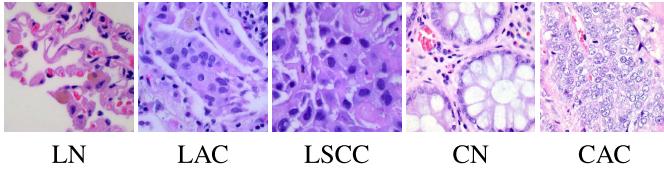


Fig. 4. LC25000 Dataset has 5 classes which includes benign lung tissues (LN), lung adenocarcinomas (LAC), lung squamous cell carcinomas (LSCC), benign colonic tissues (CN) and colon adenocarcinomas (CAC).

Dataset, histopathological images are cropped to a square size of 768×768 pixels from H&E stain WSIs of lung carcinoma and colon carcinoma. As shown in Fig. 4, LC25000 Dataset has 5 categories, including benign lung tissues (LN), lung adenocarcinomas (LAC), lung squamous cell carcinomas (LSCC), benign colonic tissues (CN), and colon adenocarcinomas (CAC). LC25000 dataset is a balanced dataset that each class has 5000 histopathological images.

V. EXPERIMENTAL RESULTS

In this section, we conduct several studies to evaluate the proposed PDBL. In the following experiments, PDBL is respectively plugged on three common classification architectures, including EfficientNet-b0 [15], ResNet50 [13] and a lightweight model ShuffLeNetV2 [34]. In Section V-A, we evaluate the effectiveness of PDBL with different proportions of training samples. In Section V-B, we test the limit of the proposed PDBL by an extremely difficult task by leveraging only 1% training samples to infer the rest of them (99%). We further compare PDBL with two state-of-the-art models in Section V-C. In Section V-D, we conduct an ablation study to verify the effectiveness and the necessity

of the pyramidal design. Next, we demonstrate the advantages of rapid domain adaptation on PDBL in Section V-E. Finally, we also show the WSI-level semantic segmentation results by stitching the patch-level classification results.

Accuracy and macro F1 score are used to evaluate the patch-level classification performance of the proposed PDBL in all the experiments. Due to the page limit, in Table II and Table III, we only demonstrate the accuracy. F1 scores can be found in the supplementary materials.

A. Effectiveness of PDBL With Different Proportions of Training Samples

In this experiment, we evaluate the effectiveness and efficiency of the proposed PDBL with different proportions of the training set in both datasets. For Kather dataset, Kather Multiclass Internal set (100k patches) and Kather Multiclass External set (7k patches) are the complete training set and test set. For LC25000 dataset, we let 60% and 40% samples as the training set and the test set. And then, the training sets of two datasets are randomly split into eight incremental subsets with the proportions of [1%, 5%, 10%, 25%, 35%, 50%, 70%, 100%], respectively. We conduct this experiment by comparing three baseline models for every CNN backbone. (1) Froze the baseline models pre-trained on ImageNet and only update the fully connected layers (FC), denoted as *Baseline + FC**. (2) PDBL directly plugged on the baseline models pre-trained by ImageNet [12], denoted as *Baseline+PDBL*. (3) Baseline models pre-trained by ImageNet fine-tuned for 50 epochs without PDBL, denoted as *Baseline**. (4) Baseline models pre-trained by ImageNet fine-tuned for 50 epochs with PDBL, denoted as *Baseline*+PDBL*. The above notations are used in all the experiments.

TABLE III

ROBUSTNESS OF PDBL IN AN EXTREME EXPERIMENT IN KATHER DATASET (TRAINING SET: 1,000 PATCHES; TEST SET: 99,000 PATCHES) AND LC25000 DATASET (TRAINING SET: 250 PATCHES; TEST SET: 24,750 PATCHES)

Models	Dataset	Accuracy					
		Fold1	Fold2	Fold3	Fold4	Fold5	Mean±SD
ShuffLeNetV2	Kather	0.16080	0.14686	0.14346	0.14352	0.14317	0.14756±0.008
		0.90529	0.91216	0.91456	0.90605	0.90610	0.90883±0.004
		0.14412	0.14964	0.15066	0.14828	0.14817	0.14817±0.002
		0.91164	0.91679	0.91831	0.91517	0.91819	0.91602±0.003
EfficientNetb0	LC25000	0.43099	0.51329	0.38517	0.47531	0.43531	0.44801±0.048
		0.95733	0.95956	0.95095	0.95301	0.95051	0.95427±0.004
		0.42461	0.43818	0.54149	0.41709	0.62816	0.48991±0.092
		0.95499	0.95414	0.95079	0.95123	0.95103	0.95244±0.002
ResNet50	Kather	0.81868	0.81378	0.81683	0.82171	0.82712	0.81962±0.005
		0.92166	0.92842	0.92677	0.92209	0.91803	0.92339±0.004
		0.90577	0.90841	0.91234	0.90911	0.91201	0.90953±0.003
		0.94894	0.95266	0.95602	0.94843	0.94926	0.95106±0.003
	LC25000	0.86974	0.88933	0.88558	0.89721	0.88630	0.88563±0.010
		0.95564	0.95681	0.95487	0.95725	0.94978	0.95487±0.003
		0.91640	0.93689	0.93345	0.94101	0.93782	0.93311±0.010
		0.96230	0.96194	0.95984	0.95960	0.96299	0.96133±0.002
	Kather	0.84314	0.85025	0.85210	0.84208	0.85559	0.84863±0.006
		0.92028	0.92207	0.92342	0.91214	0.91886	0.91935±0.004
		0.96063	0.95318	0.95742	0.95080	0.95435	0.95528±0.004
		0.96078	0.95559	0.95957	0.95207	0.95637	0.95688±0.003
	LC25000	0.89608	0.91341	0.91463	0.91240	0.91394	0.91009±0.008
		0.95091	0.95147	0.94513	0.93996	0.94404	0.94630±0.005
		0.96465	0.96756	0.95891	0.96752	0.96311	0.96435±0.004
		0.96537	0.96642	0.96040	0.96655	0.96570	0.96489±0.003

Quantitative results for three CNN backbones on two datasets are demonstrated in Table II. First, let us compare the baseline models trained by different training set proportions with and without PDBL (*Baseline** and *Baseline*+PDBL*). We can observe an overall improvement for nearly all the CNN backbones when equipping with PDBL. With the increasing training samples, the improvements became less significant for three CNN backbones. It can be observed that the degree of the improvements actually depends on three factors, the complexity of the models, the difficulties of the datasets, and the ratio of the training samples. For example, for the deeper backbone with more parameters like ResNet50, the overall improvement on a five-class dataset LC25000 is more obvious than a nine-class dataset Kather. When there are enough training samples in LC25000, say 35%, ResNet50 with PDBL gets only less than 0.001 improvement (0.99510→0.99550) but it can bring 0.018 improvement (0.97310→0.99140) for a lightweight backbone ShuffLeNetV2. It is because ResNet50 has a stronger feature representation capability than ShuffLeNetV2, which is a trade-off between computational resources and performance. Despite this, the proposed PDBL is able to steadily improve the classification performance for all the CNN backbones.

Since the proposed PDBL is a model-agnostic plug-and-play module, we are also curious about if semantic features extracted from the backbones pre-trained with ImageNet can be used to infer histopathology images. So we first froze the backbones pre-trained on ImageNet and only updated the FC layers (*Baseline+FC**). Then we directly plug PDBL on the same CNN backbones (*Baseline+PDBL*). For a lightweight backbone ShuffLeNetV2, PDBL greatly outperforms updated FC layers on both datasets, especially when fewer training

samples involve. Even in ResNet50 with 100% training samples, the accuracy of PDBL is still around 5% higher than updated FC layers. The reason is that fully connected layers (*Baseline+FC**) only consider high-level semantic features from the last convolutional layer, but PDBL (*Baseline+PDBL*) considers both high-level semantic features and low-level texture information at each stage of the CNN backbone.

According to all the results and discussion above, we can conclude that PDBL is a simple, general and effective module for any CNN classification backbone on the histopathological image classification task. When there are enough training samples (more than 50%), training the CNN backbones alone (*Baseline**) is a reliable way but PDBL can help further improve the performance. When training samples are limited (less than 25%), PDBL becomes indispensable, especially for the lightweight backbone. For extremely small training data, PDBL provides an alternative solution that uses a well-trained CNN model on ImageNet for feature extraction and to use PDBL for inference. This solution does not need to re-train the network and can greatly save computational resources and annotation efforts.

Based on this experiment, we further discuss the robustness of PDBL with an extremely small training set (only 1%) in the next experiment.

B. Robustness of PDBL With Extreme Proportions of Training and Test Sets

In the previous experiment, we observe a surprising outstanding performance of PDBL with a small training set. In this experiment, we conduct a cross-validation-like experiment, but more difficult than cross-validation, to further

discuss the robustness of the proposed PDBL. We randomly sample 1% patches (1000 for Kather; 250 for LC25000) from the training data and use the rest 99% patches (99000 for Kather; 24750 for LC25000) of the training data for testing. To alleviate the sampling bias, we repeat this experiment five times. Each repetition is regarded as a fold in this experiment. The baseline models in this experiment are totally the same as the previous one.

The same observation with the previous experiment, only updated FC layers (*Baseline+FC**) is not a reliable method when dealing with an extremely small training set. But it does not mean the features extracted from the ImageNet pre-trained backbones are not reliable. On the contrary, when plugged PDBL on the ImageNet pre-trained backbones (*Baseline+PDBL*), the performance comes back to a high standard. Experimental results show that retraining the CNN backbones (*Baseline**) is still an option for ResNet50 but not for lightweight models. Plugging PDBL on a retrained CNN backbone (*Baseline*+PDBL*) is the optimal solution. It is interesting that (*Baseline+PDBL*) performs slightly better than (*Baseline*+PDBL*) with ShuffLeNetV2 on LC25000. The reason might be that ShuffLeNetV2 fails to learn a stable model with such limited training samples (only 50 patches for each class). This extreme experiment demonstrates the potential value of the proposed PDBL in reducing the annotation efforts.

C. Comparison With State-of-the-Art Methods

In this experiment, we compare our best model *ResNet50+PDBL* with state-of-the-art methods including Kather2019 [11] and Deep MSDL [31]. For Kather Multiclass dataset, 70% KMI, 30% KMI and the entire KME are used as the training, validation and test sets, respectively. For the LC25000 dataset, we randomly divide the dataset into training, validation and test sets by the proportion of 60%, 20% and 20%. Since both PDBL and Deep MSDL are plugins, we use the same CNN backbone *ResNet50* for comparison. For the approaches without a released code, we implement them strictly following the details in the papers. We directly use the quantitative results reported in the papers (if exist).

As shown in [Table IV](#), *ResNet50+PDBL* achieves state-of-the-art performance when dealing with the complete training set on both datasets. When reducing the training samples to only 1%, none of the SOTA models achieves over 0.90 accuracy, even on the easier dataset LC25000. Our proposed PDBL achieves 0.931 for Kather Multiclass dataset, 0.961 for LC25000 dataset. It is because PDBL combines low-level and high-level features to provide more comprehensive features. And the broad learning inference is easier to find a global minimum than gradient descent when there are insufficient training samples.

D. Ablation Study of Pyramidal Design

In this experiment, an ablation study is conducted to evaluate the effectiveness of the pyramidal design of the proposed PDBL. We keep the same experimental setting of Section V-B by using 1% samples for training and 99% samples for testing

TABLE IV
COMPARISON (ACC) WITH SOTA METHODS ON KATHER MULTICLASS DATASET AND LC25000 DATASET

Method	Kather		LC25000	
	100%	1%	100%	1%
Kather2019 (VGG19) [11]	0.943	0.871	0.998	0.887
ResNet50+Deep MSDL [31]	0.959	0.894	0.998	0.880
ResNet50+PDBL (ours)	0.964	0.931	0.999	0.961

TABLE V
ABLATION OF PYRAMIDAL DESIGN (ACCURACY)

Models	Accuracy (Mean±SD)	
	Kather	LC25000
ShuffLeNetV2	+DBL	0.81173±0.013
	+PDBL	0.90883±0.004
EfficientNetb0	+DBL	0.89725±0.004
	+PDBL	0.92339±0.004
ResNet50	+DBL	0.91912±0.004
	+PDBL	0.91935±0.004

TABLE VI
ABLATION OF PYRAMIDAL DESIGN (F1 SCORE)

Models	F1 (Mean±SD)	
	Kather	LC25000
ShuffLeNetV2	+DBL	0.81464±0.013
	+PDBL	0.90947±0.004
EfficientNetb0	+DBL	0.89798±0.004
	+PDBL	0.92342±0.005
ResNet50	+DBL	0.91927±0.004
	+PDBL	0.91984±0.004

five times. Here, we only show the mean and standard deviation of the results. Note that +DBL and +PDBL represent our method without and with the image pyramid, respectively.

[Table V](#) and [Table VI](#) demonstrate the accuracy and F1 score on both datasets with only 1% training samples. We can observe a constantly stable improvement with pyramidal design for all three backbones on both datasets. According to the quantitative results, we find that the improvement of pyramidal design is more significant on the lightweight model. In Kather, PDBL can introduce around 0.09 improvement for ShuffLeNetV2, 0.03 for EfficientNetb0 and only 0.0006 for ResNet50 respectively. In fact, comparing ShuffLeNetV2 with ResNet50, ResNet50 has the stronger capacity and feature representation ability, which are great advantages when very few training samples are involved.

In addition, the same finding is also verified in this experiment that pyramidal design brings more improvement for more difficult datasets. For Kather with 9 classes, pyramidal design can introduce multi-level contextual features, which can support the feature representation for more classes.

E. Domain Adaptation Study

Domain adaptation is a crucial ability for a neural network model. In this experiment, we test if our proposed PDBL can be easily adapted from the source domain to the target domain. The same with Kather Multiclass Dataset, Zhao *et al.* [41] also released a histopathological tissue classification dataset for colorectal cancer with nine tissue types, which were collected from four different centers, including TCGA, Kather, Guangdong Provincial People's Hospital and Yunnan Cancer Center. We use the data from Guangdong Provincial People's Hospital (105k patches) as the target domain, which is divided

TABLE VII

DOMAIN ADAPTATION FROM KATHER MULTICLASS DATASET TO ZHAO *et al.* [41]. * MEANS FINE-TUNING ENTIRE CNN MODELS/FULLY CONNECTED (FC) LAYERS FOR ONE EPOCH. † MEANS UPDATING PDBL FOR ONCE

	Models	ShuffLeNetV2		EfficientNetb0		ResNet50	
		Acc	F1	Acc	F1	Acc	F1
No re-training	<i>Baseline</i>	0.77931	0.69545	0.79387	0.74014	0.83666	0.76890
	<i>Baseline+PDBL</i>	0.74612	0.67446	0.79874	0.75906	0.85536	0.80056
1%	<i>Baseline+FC*</i>	0.83666	0.80640	0.85650	0.82968	0.87006	0.85617
	<i>Baseline+PDBL†</i>	0.96336	0.95980	0.97064	0.96790	0.96848	0.96524
	<i>Baseline*</i>	0.94626	0.94032	0.93037	0.92082	0.95875	0.95416
	<i>Baseline*+PDBL†</i>	0.96275	0.95910	0.97122	0.96845	0.97183	0.96906
10%	<i>Baseline+FC*</i>	0.83702	0.80735	0.87161	0.84836	0.88172	0.86890
	<i>Baseline+PDBL†</i>	0.97952	0.97775	0.98349	0.98203	0.98190	0.98031
	<i>Baseline*</i>	0.96624	0.96295	0.96451	0.96071	0.97678	0.97448
	<i>Baseline*+PDBL†</i>	0.97930	0.97767	0.98325	0.98175	0.98454	0.98454
100%	<i>Baseline+FC*</i>	0.85275	0.82427	0.89690	0.88058	0.89821	0.88829
	<i>Baseline+PDBL†</i>	0.98482	0.98347	0.98587	0.98463	0.98444	0.98302
	<i>Baseline*</i>	0.98133	0.97952	0.98114	0.97941	0.98753	0.98641
	<i>Baseline*+PDBL†</i>	0.98590	0.98463	0.98658	0.98496	0.98987	0.98887

into a training set ($63k$ patches) and a test set ($42k$ patches) in this experiment. Let CNN backbones with and without PDBL be trained by the source domain (Kather), we first directly apply them to the target domain [41]. Next, we fine-tune the models with 100% ($63k$ patches), 10% ($6.3k$ patches) and 1% (636 patches) of the training set from the target domain, respectively. (1) The models trained on the source domain Kather Dataset (100% KMI) are denoted as *Baseline* and *Baseline+PDBL* in this study. (2) * represents fine-tuning the entire CNN models or only updating FC layers for one epoch. +PDBL† means updating PDBL by the training set of the target domain. 1%, 10%, and 100% indicate the ratios of training samples we used for fine-tuning CNN backbones and updating PDBL.

Table VII demonstrates the quantitative results of the domain adaptation study. When we directly apply baseline models (with and without PDBL) trained by Kather to Zhao *et al.* [41], the performance drastically decreases. It means that there exists a domain shift between two datasets. Then we fine-tune the models by 1%, 10%, and 100% training samples respectively, with only one epoch fine-tuning of baselines and weights updating of PDBL. *Baseline*+PDBL†* comes back to relatively high performance with the complete training set (100%). It is interesting that even we do not re-train the backbones of baseline models, *Baseline+PDBL†* can still obtain an outstanding performance by updating PDBL for once with only 1% training samples (636 patches). But *Baseline+FC** achieves only a minor improvement by updating FC layers. This observation also supports our conclusion in Section V-A that using a more stable baseline model for feature extraction and PDBL for inference is a good solution when training samples are limited. It greatly saves computational resources and annotation efforts.

F. Timing Statistics

Table VIII demonstrates the timing statistics of updating weights of PDBL versus training CNN backbones on Kather Multiclass Dataset. The time of one epoch training of CNN models is the average of 50 epochs training time. Since our proposed PDBL only needs to calculate the weights once.

TABLE VIII
AVERAGE TRAINING TIME OF BASELINE MODELS AND PDBL ON KATHER MULTICLASS DATASET (SECOND)

	CNN (1 epoch)	PDBL (Total)		
		Feature extraction	Subsequent calculation	Total
ShuffLeNetV2	157	180	33	213
EfficientNetb0	345	394	85	479
ResNet50	365	412	98	510

According to the timing statistics, the CNN backbones can get a performance boost by only spending around one epoch training time for the calculation of PDBL.

G. Semantic Segmentation for Whole Slide Images

The intention of patch-level tissue classification is to achieve semantic segmentation for whole slide images. So in this experiment, we show a colorectal WSI example of semantic segmentation by our proposed method using the WSI from the department of pathology, Guangdong Provincial People's Hospital. The model is *Baseline*+PDBL* of ResNet50 with 100% training set in this experiment.

Given a WSI in Fig. 5 (a), we first cut it into 224×224 patches under $20\times$ magnification with the resolution of 32868×39464 using sliding windows with the step size of 104 pixels. For the overlapping region, we decide the tissue class by a voting strategy. As shown in Table IX, the smaller the step size is, the more the semantic segmentation precision will be, but the more inference time it will spend. We fill the small holes of the predicted mask for better visualization with a postprocessing step. Fig. 5 (b) demonstrates the predicted semantic segmentation results. We overlay the result on the whole slide image for better visualization in Fig. 5 (c).

H. Limitation

There exists one limitation of PDBL. When dealing with large among of training samples, for example over 1 million patches, PDBL is restricted by the performance bottleneck of the hardware. Because we have to put all the patches into PDBL to learn the parameters W_{PDBL} . So the computational tractability for the large dataset is the limitation of PDBL.

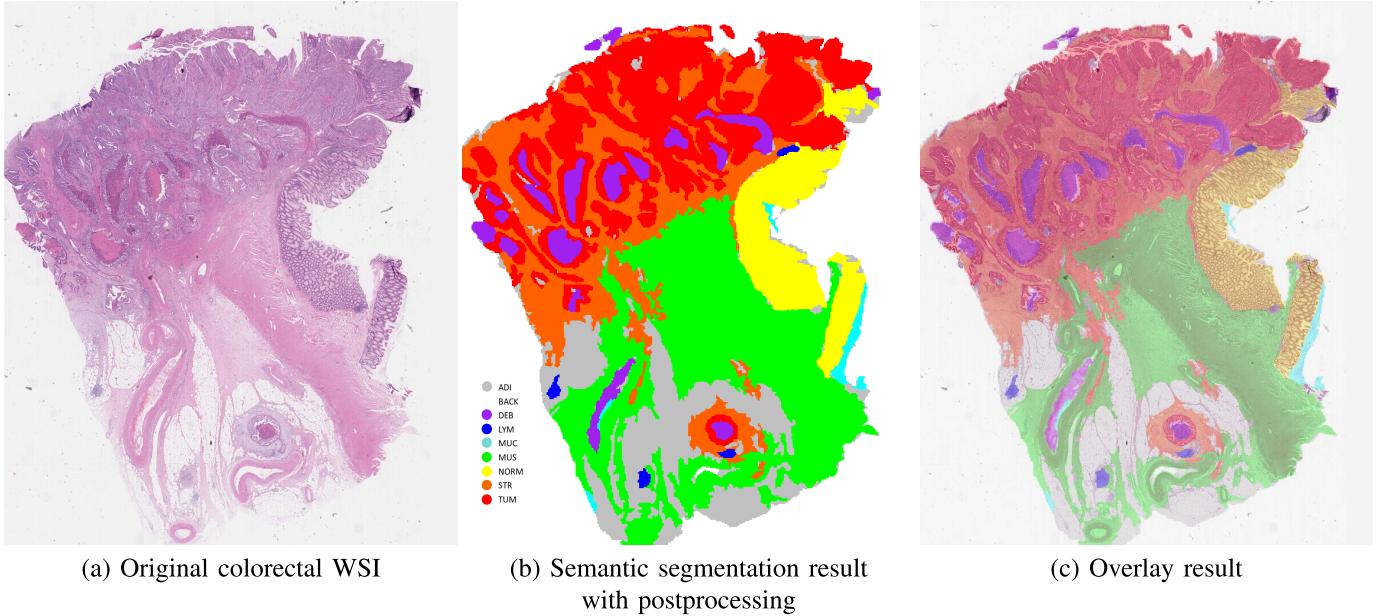


Fig. 5. Semantic segmentation results of the colorectal WSI. The original colorectal WSI (a) is cut into 224×224 patches. After predicting the class of each patch by PDBL, we tile all the patches together and form the semantic segmentation mask (b). (c) is the semantic segmentation result overlaid on the original image.

TABLE IX
INFERENCE TIME CONSUMED AT DIFFERENT STEP SIZES OF SLIDING WINDOW FOR SEMANTIC SEGMENTATION OF A WSI WITH RESOLUTION OF 32868×39464

Step size (pixel)	124	104	84
Overlap (pixel)	100	120	140
Inference time (second)	1065	1259	1590

However, the major advantage of PDBL is fast adaptation for the smaller dataset. And comprehensive experiments have already demonstrated it. In addition, whether such a large dataset is necessary is still a question since LC25000 with 25,000 patches has achieved 0.999 accuracy for five-class classification (shown in Table II). Therefore, for tissue classification task, we believe PDBL is so far capable for most of the scenarios.

VI. CONCLUSION

In this paper, we perform histopathological image classification in a new perspective by reconsidering how to make use of the deep features in order to further improve the performance of existing CNN classification backbones. Thus, we propose a lightweight plug-and-play module called Pyramidal Deep-Broad Learning for any CNN backbone without a re-training burden.

We equip this plug-and-play module on three representative CNN backbones and achieve a steady improvement of the performance using different proportions of training samples. Specifically, the proposed PDBL demonstrates good feature representation capability and inference ability when very few training samples are involved (less than 10%), especially for the lightweight models. With PDBL, we even do not have to re-train the baseline models. Such properties can greatly save computational time and annotation efforts. We also look forward to applying this plug-and-play module to more excellent

CNN backbones on the other datasets from different tumors in the future.

REFERENCES

- [1] G. Campanella *et al.*, “Clinical-grade computational pathology using weakly supervised deep learning on whole slide images,” *Nature Med.*, vol. 25, no. 8, pp. 1301–1309, Aug. 2020.
- [2] J. N. Kather *et al.*, “Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer,” *Nature Med.*, vol. 25, no. 7, pp. 1054–1056, Jul. 2019.
- [3] N. Coudray *et al.*, “Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning,” *Nat. Med.*, vol. 24, no. 10, pp. 1559–1567, Sep. 2018.
- [4] M. Gehring, M. Crispin-Ortuzar, A. G. Berman, M. O’Donovan, R. C. Fitzgerald, and F. Markowitz, “Triage-driven diagnosis of Barrett’s esophagus for early detection of esophageal adenocarcinoma using deep learning,” *Nature Med.*, vol. 27, no. 5, pp. 833–841, May 2021.
- [5] A. Binder *et al.*, “Morphological and molecular breast cancer profiling through explainable machine learning,” *Nature Mach. Intell.*, vol. 3, no. 4, pp. 355–366, Apr. 2021.
- [6] Y. Fu *et al.*, “Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis,” *Nature Cancer*, vol. 1, no. 8, pp. 800–810, Aug. 2020.
- [7] M. Y. Lu *et al.*, “AI-based pathology predicts origins for cancers of unknown primary,” *Nature*, vol. 594, no. 7861, pp. 106–110, Jun. 2021.
- [8] M. van Rijthoven, M. Balkenhol, K. Siliqi, J. van der Laak, and F. Ciompi, “HookNet: Multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images,” *Med. Image Anal.*, vol. 68, Feb. 2021, Art. no. 101890.
- [9] C. L. Srinidhi, O. Ciga, and A. L. Martel, “Deep neural network models for computational histopathology: A survey,” *Med. Image Anal.*, vol. 67, Jan. 2021, Art. no. 101813.
- [10] Y. Xue *et al.*, “Selective synthetic augmentation with HistoGAN for improved histopathology image classification,” *Med. Image Anal.*, vol. 67, Jan. 2021, Art. no. 101816.
- [11] J. N. Kather *et al.*, “Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study,” *PLOS Med.*, vol. 16, no. 1, Jan. 2019, Art. no. e1002730.
- [12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

- [14] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
- [15] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
- [16] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, "Medical image classification using synergic deep learning," *Med. Image Anal.*, vol. 54, pp. 10–19, May 2019.
- [17] L. Cai, J. Gao, and D. Zhao, "A review of the application of deep learning in medical image classification and segmentation," *Ann. Transl. Med.*, vol. 8, no. 11, p. 713, Jun. 2020.
- [18] B. Gecer, S. Aksoy, E. Mercan, L. G. Shapiro, D. L. Weaver, and J. G. Elmore, "Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks," *Pattern Recognit.*, vol. 84, pp. 345–356, Dec. 2018.
- [19] S. Wang *et al.*, "RMDL: Recalibrated multi-instance deep learning for whole slide gastric image classification," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101549.
- [20] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li, "Breast cancer multi-classification from histopathological images with structured deep learning model," *Sci. Rep.*, vol. 7, no. 1, p. 4172, 2017.
- [21] M.-J. Tsai and Y.-H. Tao, "Deep learning techniques for the classification of colorectal cancer tissue," *Electronics*, vol. 10, no. 14, p. 1662, Jul. 2021.
- [22] A. A. Borkowski, M. M. Bui, L. B. Thomas, C. P. Wilson, L. A. DeLand, and S. M. Mastorides, "Lung and colon cancer histopathological image dataset (LC25000)," 2019, *arXiv:1912.12142*.
- [23] T. C. Hollon *et al.*, "Near real-time intraoperative brain tumor diagnosis using stimulated Raman histology and deep neural networks," *Nature Med.*, vol. 26, no. 1, pp. 52–58, Jan. 2020.
- [24] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, "Patch-based convolutional neural network for whole slide tissue image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2424–2433.
- [25] Y. Xu *et al.*, "Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features," *BMC Bioinf.*, vol. 18, no. 1, pp. 1–17, 2017.
- [26] S. M. Ayyad *et al.*, "Role of AI and histopathological images in detecting prostate cancer: A survey," *Sensors*, vol. 21, no. 8, p. 2586, Apr. 2021.
- [27] H. Ni, H. Liu, K. Wang, X. Wang, X. Zhou, and Y. Qian, "WSI-Net: Branch-based and hierarchy-aware network for segmentation and classification of breast histopathological whole-slide images," in *Proc. Int. Workshop Mach. Learn. Med. Imag.*, Cham, Switzerland: Springer, 2019, pp. 36–44.
- [28] L. Rczkowski, M. Mozejko, J. Zambonelli, and E. Szczurek, "ARA: Accurate, reliable and active histopathological image classification framework with Bayesian deep learning," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Dec. 2019.
- [29] S. C. Kosaraju, J. Hao, H. M. Koh, and M. Kang, "Deep-Hipo: Multi-scale receptive field deep learning for histopathological image analysis," *Methods*, vol. 179, pp. 3–13, Jul. 2020.
- [30] J. Li *et al.*, "A multi-resolution model for histopathology image classification and localization with multiple instance learning," *Comput. Biol. Med.*, vol. 131, Apr. 2021, Art. no. 104253.
- [31] N. Hatami, M. Bilal, and N. Rajpoot, "Deep multi-resolution dictionary learning for histopathology image analysis," Dept. Comput. Sci., Univ. Warwick, Coventry, U.K., Tech. Rep., 2021, doi: [10.48550/arxiv.2104.00669](https://arxiv.org/abs/2104.00669).
- [32] H. Zhang, J. Xue, and K. Dana, "Deep TEN: Texture encoding network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2896–2905.
- [33] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1, no. 2. Cambridge, MA, USA: MIT Press, 2016.
- [34] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 116–131.
- [35] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.
- [36] C. L. P. Chen and Z. L. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- [37] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.
- [38] M. Han, S. Feng, C. L. P. Chen, M. Xu, and T. Qiu, "Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1809–1821, Sep. 2019.
- [39] M. Xu, M. Han, C. L. P. Chen, and T. Qiu, "Recurrent broad learning systems for time series prediction," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1405–1417, Apr. 2020.
- [40] S. Feng and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2018.
- [41] K. Zhao *et al.*, "Artificial intelligence quantified tumour-stroma ratio is an independent predictor for overall survival in resectable colorectal cancer," *EBioMedicine*, vol. 61, Nov. 2020, Art. no. 103054.