Efficient Classification of Very High Resolution Histopathological Images

Mohammad Iqbal Nouyed*, Gianfranco Doretto† and Donald A. Adjeroh‡
Lane Department of Computer Science and Electrical Engineering, West Virginia University
Morgantown, West Virginia, USA
Email: *monouyed@mix.wvu.edu, †gianfranco.doretto@mail.wvu.edu, ‡donald.adjeroh@mail.wvu.edu

***Abstract*—Over the years, deep learning approaches have shown significant improvement in various image understanding tasks. However, analysis of high resolution images still remains a major challenge. Apart from the huge computational resources required for such images, the large image sizes make it difficult to extract effective contextual information needed for important tasks, such as classification, segmentation, or clustering of such images. In this work, we address the challenge of high resolution image classification using a new discriminative patch selection approach. We embed our patch selection approach inside a novel classification framework, supporting potential use of different pre-trained learning models. We show results on a high resolution image dataset, namely, gigapixel whole slide tissue images for cancer tumors. We demonstrate the performance of the proposed approaches using comparative analysis with state-of-the art methods on this dataset.**

***Index Terms*—whole-slide image, WSI classification, patch-based classification, high resolution image classification, structured random sampling, GBM-LGG**

I. Introduction

With improvements in sensing, storage, communication, and computing technologies, digital images and video are now ubiquitous. Thus, the analysis of very high resolution images has found applications in various fields, from biomedicine (medical image analysis, including histopathological images) to transportation (analysis of road or rail surfaces), to energy and gas (analysis of seismic images), to land use and land transformation (analysis of remote sensing and satellite images). In this work our focus is on efficient classification of very high resolution histopathological medical images such as gigapixel whole-side images which are widely used for cancer diagnosis [7], [20]–[22]. Depending on the specific application, one basic approach to handling high resolution images is simply to treat them just like any other image, that is, apply an analysis technique directly on the high resolution image to perform the required analysis. However, given the image sizes, this may pose a significant computational problem, especially at very high resolutions, and thus a significant problem for low-resource environments.

An alternative is to perform patch-based analysis of the high resolution image by dividing the image into patches, and then applying the analysis technique on each patch. The key question becomes how we combine the results from each patch to make an overall decision about the original image. Answering this question requires a consideration of three primary challenges in patch-based image classification. (1) *Label Inconsistency:* Typically, what is available is the image label, while the actual ground truth labels for the patches are often unknown. Thus, patch labels are not necessarily the same as the true image label. (2) *Manual patch annotation:* Manual patch labeling is very time consuming, expensive, and potentially error prone. New labels will be required with each change in patch size. The large image sizes also implies that many patches may need to be considered, depending on the patch size. (3) *Errors in patch-level prediction:* Automated prediction of patch labels may not be very accurate, which can affect the final decision obtained by combining these patch-level results. Given these problems, simple direct fusion methods, for instance, majority vote, or weighted combinations may not be robust, and may not always work well. Another major issue is the huge computational resources that may be needed. Even with the patch-based approach, without a careful consideration, the analysis may still require significant overall computational resources, making it infeasible to analyze these high resolution WSIs in low-resource environments. Consider for example, the WSI images in our data set, with sizes in the range of 6000×7350 to 195215×90991 pixels. For the training set, using patch sizes of 300×300 we have a total of 39.6M patches, generated from 842 WSIs. (See Table I under experiments). This requires about 5.53 days to fine-tune a ResNet50 pre-trained model for one epoch on a single Titan RTX GPU. For 50 epochs, this would require 275 days. Efficiency in the overall processing is therefore paramount for this type of application.

In this work, we compute image-level class labels from the potentially erroneous patch-based labels in three general steps. First, we perform pre-processing on the image, and use structured sampling (when needed) to reduce the image regions to be involved in later analysis. Second, we use deep learning models to extract features, and use an information-theoretic framework to identify and eliminate non-discriminative patches. Patch re-labelling is performed (as needed) by analyzing the patch neighborhood spatial coherence. Third, by considering larger spatial regions, we perform final class-label prediction for the original high-resolution image using learning-based decision fusion on the refined patch results. Our patch-based image classification is inspired

978-1-6654-6819-0/22/$31.00 ©2022 IEEE

by the work in [7]. However, our structured patch sampling, patch-label refinement steps, and our specific attention to spatial relationships in the image contrast our approach from those of [7], where they neither refined the patch labels, nor considered regional information in their analysis. A key innovation in our work is how the patch selection algorithm is embedded in an iterative classification framework, involving the use of potentially different pre-trained learning models in extracting features from the patches. Efficiency in the approach is achieved via the steps of preprocessing, structured sampling, and the use of plug-and-play pre-trained CNN models. Beyond efficiency, this use of plug-and-play pre-trained models simplifies our approach, and also makes it more general. Performance can be improved by simply slotting in more powerful models.

II. Related Work

In recent years deep multiple instance learning based approaches such as application of patch-based classification of WSIs using convolutional neural networks (CNNs) have shown promising results for cancer classification [7], [20]–[23], [25], [26]. Researchers investigated attention based MIL using CNN for WSI classification [2], [10], [12]–[14], [24], [27], [28]. Del Amor et al. [1] proposed an inductive transfer learning framework able to perform both ROI selection and malignant prediction in spitzoid melanocytic lesions using WSIs. Yao et al. [24] proposed a method called, Deep Attention Multiple Instance Survival Learning (DeepAttnMISL) by introducing both siamese multiple instance fully connected network (MI-FCN) and attention-based multiple instance learning (MIL) pooling to efficiently learn imaging features from the WSI and then aggregate WSI-level information to patient-level. Maksoud et al. [14] presented a method for selective use of high resolution processing based on the confidence of predictions on downscaled WSIs. Lu et al. [13] presented an approach named clustering-constrained-attention multiple-instance learning (CLAM) that uses attention-based learning to identify subregions of high diagnostic value to accurately classify whole slides. Lu et al. [12] combined transfer learning and weakly supervised multitask learning to enable a single, unified predictive model to be efficiently trained on tens of thousands of gigapixel WSIs. Li et al. [10] proposed a multiple instance learning based method for WSI classification that does not require localized annotation. The method uses a novel MIL aggregator that models the relationships between the instances in a dual-stream architecture with trainable distance measurement. It is built on self-supervised contrastive learning and adopts a pyramidal fusion mechanism for multiscale WSI features.

In this work, instead of focusing on developing another novel CNN model specific to WSI classification, we investigate whether we can reuse the already existing popular CNN models for WSI classification. We construct a general multiple-instance learning based WSI classification framework by using off-the-shelf models and repurpose them for this specific classification task. We focus on providing a general framework for efficient, yet effective, CNN-based WSI classification approaches in this work.

The main contributions of this paper are three-fold: (1) We present methods for novel preprocessing and structured random sampling for efficient, yet reliable, analysis of very high resolution images; (2) We develop an iterative classification framework that can combine the power of potentially different pre-training models in classifying high resolution WSIs; (3) We introduce an information theoretic model for patch discrimination and a learning-based region-aware fusion for effective classification of high resolution histopathological images.

III. Methodology

For a given high resolution image, the key steps in our patch based classification approach are as follows: (1) efficient pre-processing and structured sampling; (2) iterative patch-based classification, using a bank of deep learning models; (3) identification and elimination of non-discriminative patches; (4) final learning-based patch decision fusion, using a region-based analysis. Below, we described the general framework, and key components in this framework.

A. Proposed Framework

Fig. 1 shows the proposed patch-based classification framework. For a given input image I, we first divide the image into non-overlapping patches. Then, we perform patch pre-processing, and structured patch sampling. The output will be a set of patches $P = \{p_1, p_2, \ldots, p_N\}$. Using fine-tuned CNN model A from a bank of pre-trained models (PM), we extract features from each patch in set P, and then perform an initial classification (block S_A) for each patch, for instance using softmax. Then we pass the softmax prediction, along with neighborhood information η and optimum set of thresholds τ^* to the patch selection (PS) stage (see below). After patch selection, we obtain the set of discriminative patches $Q = \{q_1, q_2, \ldots, q_M\}$, where $Q \subset P$. Using fine-tuned CNN model B we extract features from the selected patches in Q. We note that, the CNN models could be the same (symmetric) or different (asymmetric) at the two steps, i.e., we could have $B = A$, or $B \neq A$. Because of the neighborhood nature of the patch selection process, the overall discrimination ability of the patches could change with each set of selected patches. Thus, the two steps can be done iteratively, such that we stop the iteration when we observe little or no difference between the selected patches from one iteration to the next. The features from the final selected patches at the last iteration (using model B, the last CNN model) are then dimensionally compressed using PCA and passed to a classification module (block S_B), such as a Random Forest (RF) or Support Vector Machines (SVM) for final patch level classification. The image-level classification is then performed through a region-based analysis on the individual patch results, the results of which are then fed to a learning-based region-aware fusion scheme. (This is the LRF block in the figure). The outcome of

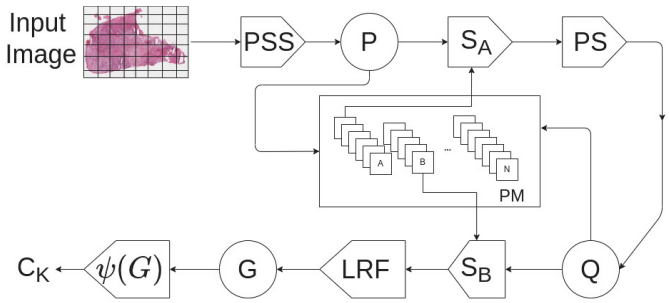

Fig. 1. Workflow diagram for proposed patch-based classification framework. Notations: *PSS: preprocessing and structured sampling*; *P: output patches after preprocessing and sampling*; *PM: bank of pretrained CNN models*, S_A: *Initial patch classification*; S_B: *second patch classification*; *PS: patch selection using information-theoretic identification of non-discriminative patches*; *Q: output patches after discrimanative patch selection*; *LRF: learning-based and region-based patch decision fusion*; *G: learned region-based attributes*; $\psi()$: *final image-level classification*; C_K: *predicted output class*

the LRF block is then used for final image-level classification of the input WSI.

B. Efficient processing

1) Preprocessing: Usually whole-slide histopathological images have a lot of blank regions surrounding the tissue image. So when patches are extracted, some of the patches are may not have texture or may be marginally covering part of the tissue sample.

Thus, during our structured random sampling step (discussed below), we have to make sure that we only select "non-blank" patches. In order to do this, we at first perform Canny edge detection on the patch, then label the connected regions. By definition, two pixels are considered to be connected when they are neighbors and have the same value. In an image patch, they can be neighbors either in a 1- or 2-connected sense. For our case, a single orthogonal hop is used to consider a pixel as a neighbor. For each identified region within the patch we calculate the area and sum these up to get the total area of the connected regions within the patch. We calculate this "area of connected regions" (denoted A_{total}), and create a frequency distribution of A_{total}. Finally, we check each bin of the distribution and decide on an appropriate threshold which we use to filter out the blank (or mostly blank) patches.

2) Structured sampling: Given the very high resolution of the WSIs (images in the TCGA cancer datasets range from $6,000 \times 7,350$ to $195,215 \times 90,991$), extracting relatively small-sized patches creates a huge number of such patches for each image (even after pre-processing). This makes the model training computationally very expensive. Thus, we needed to develop a sampling strategy that could effectively select a smaller number of patches to represent the image without losing too much information about the spatial correlation in the original high resolution image. This requires us to determine an effective number of patches, and appropriate patch size, both of which will depend on the original image size. To decide on the patch sample size of a whole-slide image, we used two approaches. The first is a simple approach using a fixed number of patches, for instance, 100 patches per image (ppi). The second is an adaptive approach depending on the image size, using the Cochran's formula [3] for sample size calculation with a known population size:

$$n = \frac{\frac{z^2 p(1-p)}{e^2}}{1 + (\frac{z^2 p(1-p)}{e^2 N})} \tag{1}$$

where n is the sample size (number of patches), e is the desired level of precision (i.e. the margin of error), p is the (estimated) proportion of the population which has the attribute in question. The value of z is found in a Z table. N is the known population size. Armed with the number of patches, we still have to decide on the most effective way to determine exactly *which* patches will be selected to make up the number. Given the unconstrained nature of the shape of a tissue sample, and the existence of blank and non-informative patches, to properly sample the slice we have to develop a strategy that takes more samples from regions where informative patches are available and avoid taking samples from non-informative regions. A stratified random sampling [8] is too costly because it will take uniform samples from each strata regardless of the utility of the patches for classification. Thus, we propose to use an adaptive approach, where we divide the image into grids or cells of size $\frac{r}{\sqrt{n}} \times \frac{c}{\sqrt{n}}$, where, n is the computed sample size from Equation (1), and r and c denote the respective number of rows and columns in the image. Initially, for each grid, we randomly select one non-blank patch whose centroid lies within the grid. Given the preprocessing step, it is possible to find a grid that contains only blank patches. Thus, if after a single iteration over the image we do not reach a total of n patches (i.e., some grids had only blank patches), then we consider only cells that have produced a non-blank patch to get the remaining patches. This structured sampling strategy provides two advantages: (1) it ensures a more uniform spatial distribution of the randomly selected patches over the entire image; (2) it avoids selection of the blank patches (as determined by the threshold on A_{total}), which thus ensures that selected patches will be from informative regions in the high resolution image; (3) maintains the general spatial coherence among the selected patches. Algorithm 1 (Fig. 2) captures this adaptive structured sampling strategy.

C. Iterative patch selection and classification

1) Discriminative patch selection and patch relabeling: A fundamental step in our approach is the patch selection stage (denoted as **PS** in Fig. 1). Each patch provides a contribution to the decision on the class of the input image, for instance, using a simple majority vote on the predicted patch labels. Let $P(U) = \{p_U(1), p_U(2), \ldots, p_U(c), \ldots, p_U(|C|)\}$ denote the predicted class probability distribution for patch U, where C is the number of image classes. For a given image patch U, the class probabilities $P(U) = \{p_U(c)\}_{c \in C}$ are obtained from the output of softmax. Traditionally, the class with the highest probability is chosen as the predicted class for the patch. However, this simple decision can lead to misclassifications,

Require: $grid\_size, patch\_size, img\_size, patch\_list$
Ensure: $sampled\_patch\_list$
1: $ppi \leftarrow get\_sample\_size(patch\_size, img\_size)$
2: $ppc \leftarrow ppi/grid\_size$
3: $cell\_positions \leftarrow get\_cell\_positions(img\_size, grid\_size)$
4: $grpd\_patches \leftarrow \ldots$
5: $grp\_patches\_by\_cell(patch\_list, cell\_positions)$
6: **for** $cell\_patches \leftarrow grpd\_patches$ **do**
7: **while** $patch\_count < ppc$ **do**
8: $id \leftarrow random\_sample(cell\_patches)$
9: $is\_blank \leftarrow blank\_check(cell\_patch[id])$
10: **if** $not\ is\_blank$ **then**
11: $sampled\_patch\_list.append(cell\_patch[id])$
12: **end if**
13: **end while**
14: **end for**{If ppi is not met, we repeat the process, by uniformly sampling from cells that have produced ppc samples, thus ignoring cells with mostly blank patches.}

Fig. 2. Algorithm 1: Algorithm for structured random sampling

especially, if the class probabilities are not well separated. Further, because decisions are patch-based, it becomes more difficult to train the system on the basis of global labels for the original high resolution image. Thus, before we use the predicted label for patch U in the overall decision, we check if the classification result for patch U is discriminative enough to be used. First, we compute the class entropy for patch U:

$$H(U) = \sum_{c} p_U(c) \log\left(\frac{1}{p_U(c)}\right) \tag{2}$$

We say the patch is complex if $(\underset{c\in C}{max}\{p_U(c)\} - \underset{c\in C}{max}2\{p_U(c)\}) \leq \tau_p$ or if $H(U) \geq \tau_H$. Here, $max2$ is a function that returns the second largest value in a set, and τ_P and τ_H are thresholds. If the patch is not complex, we simply assign it the class label: $L(U) = c^* = \underset{c\in C}{argmax}\{p_U(c)\}$. For a complex patch, we then consider its spatial coherence with its neighborhood. First, we perform the above patch complexity check on each patch in the neighborhood. If less than half of the patches in the neighborhood are found to be complex, we proceed with the spatial coherence analysis. If not, (i.e., more than half of the patches in the neighborhood are complex), we say the patch is non-discriminative, and thus do not use it further for classification of the image. For neighborhood coherence analysis, we use the Jensen-Shannon divergence [4] between U and its neighbors. For two probability distributions A and B, the Jensen-Shannon divergence is given by:

$$JSD(A, B) = \frac{1}{2}D(A||Q) + \frac{1}{2}D(B||Q) \tag{3}$$

where $Q = \frac{1}{2}(A + B)$, and $D(A||Q)$ is the Kullback-Leibler (KL) divergence [4] between two distributions, given by:

$$D(A||Q) = \sum_{c=1}^{|C|} A(c) \log\left(\frac{A(c)}{Q(c)}\right) \tag{4}$$

Then, for the patch U, we compute the mean and standard deviation of the Jensen-Shannon divergence between U and every other patch in its neighborhood, viz:

$$\mu_U = \frac{1}{|\mathcal{N}|}\sum_{V\in\mathcal{N}} JSD(P(U), P(V)) \tag{5}$$

$$\sigma_U = \sqrt{\frac{1}{|\mathcal{N}|}\sum_{V\in\mathcal{N}} (\mu_U - JSD(P(U), P(V)))^2} \tag{6}$$

where $\mathcal{N}$ denotes the patch neighborhood for patch U. If $\mu_U \leq \tau_\mu$, we say that patch U is coherent with its neighbors. Similarly, if $\sigma_U \leq \tau_\sigma$, it means that the patch neighborhood for patch U is coherent (i.e., all the neighbors have similar differences with U). Here again, τ_μ and τ_σ are two thresholds. Thus, for the complex patch, if its neighborhood is coherent, we assign it the class label ($L(U)$) using the dominant class in its neighborhood. Similar to patch complexity, we determine the dominant class as follows:

$$v^* = \underset{V\in\mathcal{N}}{argmax}\{p(L(V))\} \tag{7}$$

$$\text{if } \underset{V\in\mathcal{N}}{max}\{p(L(V))\} - \underset{V\in\mathcal{N}}{max}2\{p(L(V))\} \geq \tau_L \tag{8}$$

where $p(L(V))$ denotes the probability of label $L(V)$ (the predicted label for patch V) in the neighborhood, and τ_L is a threshold. Then, we assign patch U the label of this dominant class in its neighborhood: $L(U) = L(v*)$. If the neighborhood is not coherent, or there is no dominant class label, we can't rely on information from the neighborhood. Thus, we say the patch is non-discriminative, and remove it from further analysis. For patch selection, a set of thresholds (τ) – defined above – are determined during training. At first the thresholds are initialized based on empirical observations. Then, the selection of optimal set of thresholds $\tau* = \{\tau_p, \tau_H, \tau_\mu, \tau_\sigma, \tau_L\}$ is based on patch level classification performance.

2) Learning-based image level fusion: The final step is image-level label prediction for the high resolution WSI using the results of the above patch-level analysis (denoted as **LRF** in Fig. 1). As noted previously, one way to do this will be to perform a simple majority vote, using the patch level classification results. A key observation is that, in addition to the individual label for each patch, the distribution of patch labels within a neighborhood might play a major role in determining the actual label of the larger image. For instance, errors in the patch-based labels could be corrected based on information from other nearby patches. Given the size of the high resolution images, rather than simple majority vote, we consider the nearby patches within regions in the image, and then use a learning-based decision fusion to predict the class of the original high resolution image. For image regions, we simply divide the image into spatial regions, for instance 9 regions (by dividing the original image into 3 horizontal $\times$ 3 vertical regions). The number and size of a region will typically depend on the size of the original high resolution image. Although image regions also capture some spatial relations in the image, we note that image regions are generally larger than patch neighborhoods, as used previously above. Let

$P(R_i) = \{p_{R_i}(c)\}_{c \in C}$ denote the predicted class probability distribution for the i-th region, R_i. For each region R_i, we compute the distribution $P(R_i)$ using the predicted class label for each surviving discriminative patch in the region. Then, using $P(R) = \{P(R_i)\}_{i=1,2,...,|R|}$ the set of class distributions from all the regions, we train a simple classifier, such as a support vector machine (SVM), random forest, or simple multilayer perceptron (MLP) for final label predication for the original image.

IV. Experiments and Results

A. Setup and Implementation

Datasets: We use a combined subset of the Glioblastoma (GBM) and Low-Grade Glioma (LGG) cancer dataset from The Cancer Genome Atlas (TCGA) [19]. This subset was also used by Hou et al. [7]. It contains 842 WSIs in training set and 222 in test set[1]. The train-test are divided with non-overlapping patient id. A brief summary of the dataset is shown in Table I. This data subset has six subtypes of Glioma, namely, Glioblastoma (GBM), Oligodendroglioma (OD), Oligoastrocytoma (OA), Diffuse astrocytoma (DA), Anaplastic astrocytoma (AA), Anaplastic oligodendroglioma (AO). Of these six subtypes, five are classified as Low-Grade Glioma (LGG), namely, OD,OA,DA,AA, and AO. More detailed description of these cancer subtypes are presented in [7]. The combination of GBM, LGG, and large image sizes makes this dataset very unique and not investigated much by other researchers. We wanted to further investigate the challenge of using this high-resolution dataset.

TABLE I
INFO ON TCGA GBM-LGG DATASET USED IN OUR EXPERIMENTS

	Train	Test	Total
No. of WSIs	842	222	1064
Avg. WSI res.	69953×51058	67322×47298	-
Max. WSI res.	195215×90991	181941×86057	-
Min. WSI res.	6000×7350	12000×16391	-
No. of patches	39.6M	9M	48.6M
Avg. PPI	47047.9	41544.0	-
Max PPI	196950	173316	-
Min PPI	456	2106	-
Est. train time (1 epoch)[a]	5.53 days	-	-

[a]Estimated time is based on training time calculated for 100ppi trainset shown in Table VIII

Structured Random Sampling w/ blank detection: For each patch we compute A_{total}, the total area of connected regions. Using A_{total} we place the image patches into 20 bins. Fig. 3 shows sample patches from some of the bins, and how A_{total} provides an effective indicator of blank or non-blank patches. Cases with A_{total} = 0 or A_{total} =1 accounted for about 25% of the patches. Figure 4 shows the distribution of A_{total} for a sample WSI. We can see the high frequency of "blank patches" in bin 0. We used 300×300 size patches, since the pre-trained CNN models used for our experiments accepted input images of size 224×244 or 299×299. Using Cochran's formula

[1]We used 217 test images as we found 5 images missing in the downloaded TCGA manifest.

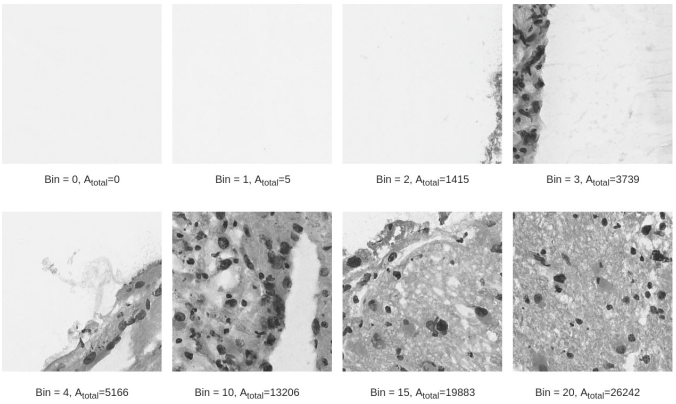

Fig. 3. Sample patches from eight bins of the A_{total} histogram. A_{total} provides a good indicator of potential blank patches. Lower values indicate more likelihood to be blank.

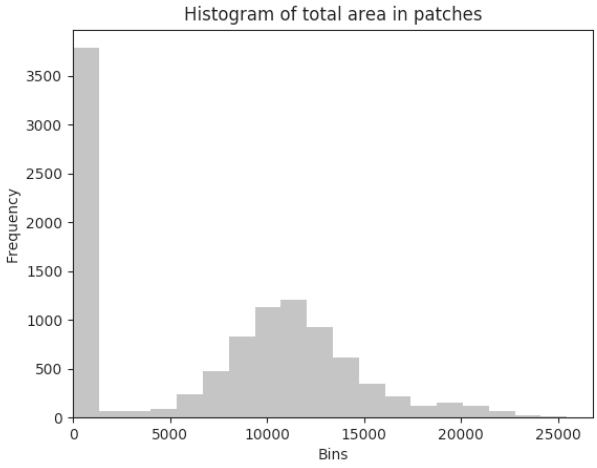

Fig. 4. Frequency distribution of all patches of a sample WSI based on "total area of connected regions", shown in 20 bins. A high number of patches with 0 total area can be observed.

(Eqn. (1)) we compute the number of patches needed for each image. Since we took the nearest 100th value larger than the calculated number of patches sample, the actual number of patches ranged between 100 to 400. After this step, the result of the structured random sampling with blank detection reduced the training set from 39,614,333 to 329,059 patches (0.8% of the original training set) and test set from 9,015,053 to 82,970 patches (0.9% of the original test set). A 5×5 grid is used to perform the structured random sampling.

Fine-tuning CNNs: We experimented with a bank of 4 popular CNN models that have been pre-trained on the ImageNet dataset, namely, Resnet50 [5], InceptionV3 [18], VGG16 [17], and VGG19BN [17]. These are respectively finetuned using the input dataset. We used data augmentation to further increase the training set. We use pre-trained models provided by the Pytorch [16] library. Stochastic gradient descent with cross-entropy loss is used to fine-tune. Learning rate was initialized at 0.001 and reduced by 0.1 when validation loss plateaus; momentum was set to 0.9. We ran the fine-tuning for 50 epochs. Dataset was divided into train and test sets using the 80-20 split. For data augmentation, at first image is resized to

fit the pre-trained model, then affine transformation is applied such as, rotation (within 15 degrees), translation (within 0.1 in both horizontal and vertical directions), scaling (0.9 to 1.1) and shear (up to 1.1). Along with this, horizontal flip was randomly applied and finally patch was normalized to 0.5 mean and 0.5 standard deviation in all 3-channels. We randomly cropped the required CNN input size from 300×300 patches during training, and applied center crop at testing stage.

Building the neighborhood patch matrix: To determine the neighboring patches we use the centroid of the patches and their Euclidean distance to each other. Usually a patch can have 8 adjacent neighbors, but, because of the possible elimination of blank patches, these neighbors can be further away. For this, reason we use an incremental search radius from the centroid of the patch under consideration to search the neighborhood patches. Initially, search radius $r = Cx\sqrt{2}$ where x = height or width of the square patch and C is a multiplicative factor. If a candidate neighbor patch centroid distance $d \leq r$, it is considered as neighboring patch. We iterate through all the patches of the image until we get the 8 neighborhood patches, if not we increase the value of C by 1 (initially $C = 1$), which increases the search radius, and then we search again. We keep incrementing search radius until $C = \sqrt{I_w^2 + I_h^2}$ or at least 8 neighbors are found, where I is the WSI in consideration, I_w and I_h denotes its width and height, respectively.

Optimal threshold search for patch selection: For all patches in the training set, the entropy $H(U)$, probability difference, patch complexity, mean and standard deviation of Jensen-Shannon divergence are calculated and stored in individual lists. Then entropy threshold τ_H is determine from the 90th percentile of the entropy. Similarity, τ_P is selected from the 10th percentile. Mean and standard deviation threshold of Jensen-Shannon divergence, τ_μ and τ_σ, and patch complexity (dominant patch) threshold τ_L, is determined from a set of 5 percentile values $\{10, 25, 50, 75, 90\}$. Based on these threshold variations, $1 \times 1 \times 5 \times 5 \times 5 = 125$ different patch selection combinations are generated. Among them, the top performing thresholds are determined empirically, by their performance in patch-level classification using the training datasets.

B. Results

1) Overall Result: Table II shows comparative assessment of our methods with the state-of-the-art. In the tables, "rgn" is used to denote region. Though there are quite some recent work on binary classification on this dataset, we could not find a lot of recent work on multi-class classification using this dataset. However, because of the unique classification challenge presented by this dataset (specially LGG cancer types), we chose to work with this database. We can observe that for binary classification our proposed method produced the best accuracy of 98%. For 6-class classification we obtained competitive mAP and top-1 accuracy compared to EM-CNN-LR [7]. These are very significant results, especially in the context of limited resource environments (our methods are based on just 0.8% of the training set).

TABLE II
PERFORMANCE ON GBM-LGG DATASET. NOTE THAT OUR PROPOSED METHODS USED ONLY 0.8% OF ORIGINAL TRAINING DATA.

Method	Top-1 acc	mAP
	GBM vs LGG	
FocAtt-MIL-DN [9]	0.89	-
ResNet50+SAM+Contrast(SMILE) [11][a]	0.88	-
CNN-DS [6]	0.91	-
EM-CNN-LR [7]	0.97	-
Spatial & morphological filters [15]	0.85	-
Chance	0.50	-
ResNet50+VGG16+9rgn, (ours)	**0.98**	-
	6-class classification	
EM-CNN-LR [7]	**0.77**	**0.85**
VGG16+VGG16+9rgn,100ppi (ours)	0.74	0.84
ResNet50+VGG16+9rgn, (ours)	0.73	0.82

[a]Method in [11] used 3 classes (GBM, A and O) instead of two.

TABLE III
GBM-LGG DATASET CLASSIFICATION RESULTS (TOP-1 ACCURACY) AT PIPELINE STAGE PSS+S_A (I.E. BEFORE PATCH SELECTION).

CNN Model	Patch level	Image level
Resnet50	0.679	0.694
InceptionV3	0.637	0.657
VGG16	0.679	0.718
VGG19BN	0.678	0.694

TABLE IV
GBM-LGG DATASET CLASSIFICATION RESULTS (TOP-1 ACCURACY) AT PIPELINE STAGE PSS+S_A+PS+S_B (I.E. AFTER PATCH SELECTION, BUT BEFORE REGION-BASED ANALYSIS). FOR BREVITY, ONLY RESULTS FROM THE TOP PERFORMING MODELS ARE SHOWN

CNN Model A	CNN Model B	Patch-level	Image-level
Resnet50	Resnet50	0.679	0.708
InceptionV3	InceptionV3	0.637	0.671
VGG16	VGG16	0.681	0.732
VGG19bn	VGG19bn	0.678	0.718
ResNet50	VGG16	0.679	0.718
ResNet50	VGG19bn	0.679	0.713
Inceptionv3	VGG16	0.680	0.727
VGG19bn	VGG16	0.680	0.727

TABLE V
CLASSIFICATION RESULTS (TOP-1 ACCURACY) AFTER COMPLETE PIPELINE RUN OF THE FRAMEWORK: PSS+S_A+PS+S_B+LRF.

CNN Model A	CNN Model B	Region type	Accuracy
Resnet50	VGG16	1-region	0.722
Resnet50	VGG16	4-region	0.727
Resnet50	VGG16	9-region	0.713
VGG16	VGG16	1-region	0.727

2) Ablation Study: We perform ablation study in two parts, (1) effect of the framework before patch selection, this focuses on $PSS + S_A$ stage of the pipeline (see Fig. 1), and (2) effect of the framework after patch selection, this focuses on $PSS + S_A + PS + S_B$ stage of the pipeline. Tables III, IV and V show the results of our ablation studies where we inspect the classification performance at different stages of the proposed framework.

Effect of $\mathbf{PSS + S_A}$: Table III shows the baseline performance right after structured random sampling is done and models are fine-tuned using the sampled dataset. Image level

results are obtained using majority vote. Random Forest classifier was trained using the CNN features to get the results. This step of PSS is key to efficiency as it resulted in a reduction of the required training data to just 0.8% of the original training set (See Table VI, 400ppi). This allowed us to fine tune the CNN model in 50 epochs where each epoch took about $3,276$ seconds (as compared to 5.53 days – see Table VIII). We will compare this result with performance after patch selection.

Effect of $\mathbf{PSS + S_A + PS + S_B}$: Table IV shows the result further down the framework where patch selection is applied, but without the decision fusion step. We observe that classification has improved, at both patch level and image level. This indicates that discriminative patch selection has further improved the results. Also, notice that using different CNNs after patch selection (asymmetric case) can be a useful technique, for instance, using ResNet50+VGG16 and VGG19bn+VGG16 in the pipeline gives us close to the best result.

Effect of $\mathbf{PSS + S_A + PS + S_B + LRF}$: Finally, Table V shows the entire run of the framework with the result of learning-based decision fusion. We can observe the improvement both in patch and image level classification at each stage, which justifies the contribution of that specific module to the framework. The use of different combination of off-the-shelf models can produce classification accuracy at different rates, but the overall improvement in accuracy is consistent.

3) Computational requirements: A major consideration in this work is computational efficiency – given the huge data sizes involved in analyzing WSIs. The challenge is to develop effective deep learning approaches that can perform well in terms of analysis results, while still remaining cost effective, with low resource requirements. As noted earlier, our results above were achieved using just about 0.8% sampled patches from the entire training set (when adaptive (structured) sampling strategy is applied). Our training set has 329,059 patches, where the original training set has 39,614,333 patches. Below, we check whether this sampled dataset provides an effective representation of the original dataset, and analyze the required space and time (execution time) for the proposed method. We vary the sample size by increasing or decreasing the Cochran factor (Eqn. (1)), by multiplying with a constant value. Table VI shows the sizes for the training sets created by applying this method. After that, we run our framework using ResNet50 at S_A and VGG16 at S_B modules, Table VII shows the results. We can notice that as the sample size increases the classification accuracy increases, but after about 400ppi it stops increasing. This indicates that a good representative set of samples have been obtained, at 400ppi.

Table VIII provides information on the time it takes to run each of the modules. We use a single Nvidia Titan RTX GPU (on a 40 core Intel (R) Xeon 2.29GHz machine) to run all the fine-tunings and feature extractions. This table provides a comparative measure of how fast the time requirement increases as the sample size increases, which indicates the necessity of this framework when computational resources are limited, or we need fast execution. Image as unit means the time it requires to process all the patches of an image. Threshold combination time is the processing time required for one out of 125 combinations in patch selection module. Image feature construction/threshold combination, is the time required to construct an image feature using the region based fusion method. Notice that, apart from the PSS module, all other steps are measurement for training phase. We can observe that, the major bottleneck is the CNN training time followed by LRF step. Table IX provides an analysis on memory requirement of the framework as we handle a larger set of patches per image in train and test set. Each 300×300 patch size is considered 182kb on average. Memory requirement is measured (in bytes) at each section for each unit. Image as unit means the total memory it requires to process all the patches of an image. The total size of parameters used and buffer size is used to calculate memory for the CNNs. We use the size of the input and output variables and threshold combinations to calculate the memory requirement at PS and LRF stages.

TABLE VI
TRAINING SET SIZES FOR THE DIFFERENT SAMPLE SIZES PER WSI.

Sample size	Cochran factor	percentage of original trainset	No. of patches (train)[a]
100ppi	.25	0.2	82,719
200ppi	.5	0.4	165,442
400ppi	1	0.8	329,059
800ppi	2	1.6	643,318

[a]Note that, WSIs do not always produce the same number of samples because of blank patch removal (PSS stage).

TABLE VII
TOP-1 ACCURACY FOR RESNET50+VGG16+X-RGN METHODS USING 4 DIFFERENT SAMPLE SIZES ON A COMMON TESTSET (400PPI)

LRF	100ppi	200ppi	400ppi	800ppi
1-rgn	0.64	0.66	0.72	0.68
4-rgn	0.65	0.66	0.73	0.68
9-rgn	0.65	0.67	0.71	0.68

TABLE VIII
TIME MEASURED (IN SECONDS) AT EACH STAGE OF THE FRAMEWORK.

Module	Unit of measure	100ppi	200ppi	400ppi	800ppi
PSS	image	34.0	46.3	54.1	74.5
S_A (RN50)	1 epoch	1015	1802	3276	6211.3
PS	thresh. combi.	1.5	3.3	6.6	13.8
S_B (VGG16)	1 epoch	1225	2107	4196	7599
LRF(1-rgn)	image feat. const. / thresh comb.	4.6	9.1	20.5	36.3
LRF(4-rgn)	image feat. const. / thresh comb.	44.3	168.2	771.2	2488.8
LRF(9-rgn)	image feat. const. / thresh comb.	43.6	161.8	758.2	2488.3

V. CONCLUSION

We presented a general framework for patch-based high resolution image classification, specifically for whole-slide images, that has shown promising results. Our proposed framework adopts a domain specific spatial sampling strategy with blank patch detection, and identifies and selects discriminative patches using an information-theoretic approach. It then embeds the selected patches in a novel classification framework which supports feature extraction from a bank of potentially different fine-tuned learning models. Using this approach we studied four different off-the-shelf CNN models

TABLE IX
MEMORY REQUIREMENT (IN MB) AT EACH STAGES OF THE FRAMEWORK.

Module	Unit of measure	100ppi	200ppi	400ppi	800ppi
PSS	image	18.2	36.4	72.8	145.6
S_A (RN50)	param size + buffer size[a]	97.7	97.7	97.7	97.7
PS	input + output, thresh. comb.	11.7	23.4	45.5	67.5
S_B (VGG16)	param size + buffer size	527.8	527.8	527.8	527.8
LRF	input + output, thresh. comb.	3.2	6.4	12.6	25.2

[a]Note that, memory usage of CNNs are computed based on parameter size and buffer size when a single patch is taken as input. Size of feature vector is ignored which is small compared to the parameter size.

in the framework and were able to classify a cancer glioma dataset, which showed better or competitive results with state-of-the art methods. We have achieved comparable results just by using a tiny fraction of the dataset (0.8% of the original training set). The use of off-the-shelf pre-trained models makes our framework easily extensible to use more sophisticated models which can improve the results further. Our future work includes investigating the effectiveness of this framework for other types of deep learning models such as transformers and graph convolutional networks.

ACKNOWLEDGMENT

This work is supported in part by grants from the US National Science Foundation (Award 1761792, 1920920, 212587)

REFERENCES

[1] "An attention-based weakly supervised framework for spitzoid melanocytic lesion diagnosis in whole slide images," *Artificial Intelligence in Medicine*, vol. 121, p. 102197, 2021.

[2] C.-L. Chen, C.-C. Chen, W.-H. Yu, S.-H. Chen, Y.-C. Chang, T.-I. Hsu, M. Hsiao, C.-Y. Yeh, and C.-Y. Chen, "An annotation-free whole-slide training approach to pathological classification of lung cancer types using deep learning," *Nature Communications*, vol. 12, no. 1, pp. 1–13, 2021.

[3] W. G. Cochran, *Sampling Techniques (3rd. ed.)*. John Wiley & Sons, 1977.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory (2. ed.)*. Wiley, 2006. [Online]. Available: http://www.elementsofinformationtheory.com/

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[6] S. Hemati, S. Kalra, C. Meaney, M. Babaie, A. Ghodsi, and H. Tizhoosh, "CNN and deep sets for end-to-end whole slide image representation learning," in *Medical Imaging with Deep Learning*, 2021.

[7] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] S. K. Thompson, *Sampling, Third Edition*. Canada: Wiley, 2012.

[9] S. Kalra, M. Adnan, S. Hemati, T. Dehkharghanian, S. Rahnamayan, and H. R. Tizhoosh, "Pay attention with focus: A novel learning scheme for classification of whole slide images," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds. Cham: Springer International Publishing, 2021, pp. 350–359.

[10] B. Li, Y. Li, and K. W. Eliceiri, "Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 318–14 328.

[11] M. Lu, Y. Pan, D. Nie, F. Shi, F. Liu, Y. Xia, and D. Shen, "SMILE: Sparse-attention based multiple instance contrastive learning for glioma sub-type classification using pathological image," in *COMPAY 2021: The third MICCAI Workshop on Computational Pathology*, 2021. [Online]. Available: https://openreview.net/forum?id=12-kKzWtSL0

[12] M. Y. Lu, T. Y. Chen, D. F. Williamson, M. Zhao, M. Shady, J. Lipkova, and F. Mahmood, "AI-based pathology predicts origins for cancers of unknown primary," *Nature*, vol. 594, no. 7861, pp. 106–110, 2021.

[13] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data-efficient and weakly supervised computational pathology on whole-slide images," *Nature Biomedical Engineering*, vol. 5, no. 6, pp. 555–570, 2021.

[14] S. Maksoud, K. Zhao, P. Hobson, A. Jennings, and B. C. Lovell, "SOS: Selective objective switch for rapid immunofluorescence whole slide image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3862–3871.

[15] H. S. Mousavi, V. Monga, G. Rao, and A. U. Rao, "Automated discrimination of lower and higher grade gliomas based on histopathological image analysis," *Journal of Pathology Informatics*, vol. 6, 2015.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," 2019.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[19] TCGA, "The Cancer Genome Atlas (TCGA) Program. https://tcga-data.nci.nih.gov/tcga/," 2021. [Online]. Available: https://tcga-data.nci.nih.gov/tcga/

[20] D. Tellez, J. van der Laak, and F. Ciompi, "Gigapixel whole-slide image classification using unsupervised image compression and contrastive training," *Medical Imaging with Deep Learning (MIDL'2018) Conference Track*, 2018.

[21] Q. D. Vu, S. Graham, T. Kurc, M. N. N. To, M. Shaban, T. Qaiser, N. A. Koohbanani, S. A. Khurram, J. Kalpathy-Cramer, T. Zhao *et al.*, "Methods for segmentation and classification of digital microscopy tissue images," *Frontiers in Bioengineering and Biotechnology*, vol. 7, 2019.

[22] X. Wang, H. Chen, C. Gan, H. Lin, Q. Dou, E. Tsougenis, Q. Huang, M. Cai, and P.-A. Heng, "Weakly supervised deep learning for whole slide lung cancer image analysis," *IEEE Transactions on Cybernetics*, 2019.

[23] Y. Xu, Z. Jia, Y. Ai, F. Zhang, M. Lai, and E. I. Chang, "Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 947–951.

[24] J. Yao, X. Zhu, J. Jonnagaddala, N. Hawkins, and J. Huang, "Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks," *Medical Image Analysis*, vol. 65, p. 101789, 2020.

[25] J. Ye, Y. Luo, C. Zhu, F. Liu, and Y. Zhang, "Breast cancer image classification on wsi with spatial correlations," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 1219–1223.

[26] C. Zhang, Y. Song, D. Zhang, S. Liu, M. Chen, and W. Cai, "Whole slide image classification via iterative patch labelling," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1408–1412.

[27] J. Zhang, K. Ma, J. Van Arnam, R. Gupta, J. Saltz, M. Vakalopoulou, and D. Samaras, "A joint spatial and magnification based attention framework for large scale histopathology classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3776–3784.

[28] Y. Zhao, F. Yang, Y. Fang, H. Liu, N. Zhou, J. Zhang, J. Sun, S. Yang, B. Menze, X. Fan *et al.*, "Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4837–4846.