Computerized Classification of Prostate Cancer Gleason Scores from Whole Slide Images

Hongming Xu, Sunho Park, and Tae Hyun Hwang

Abstract—Histological Gleason grading of tumor patterns is one of the most powerful prognostic predictors in prostate cancer. However, manual analysis and grading performed by pathologists are typically subjective and time-consuming. In this paper, we present an automatic technique for Gleason grading of prostate cancer from H&E stained whole slide pathology images using a set of novel completed and statistical local binary pattern (CSLBP) descriptors. First the technique divides the whole slide image (WSI) into a set of small image tiles, where salient tumor tiles with high nuclei densities are selected for analysis. The CSLBP texture features that encode pixel intensity variations from circularly surrounding neighborhoods are extracted from salient image tiles to characterize different Gleason patterns. Finally, the CSLBP texture features computed from all tiles are integrated and utilized by the multi-class support vector machine (SVM) that assigns patient slides with different Gleason scores such as 6, 7 or $\geq$8. Experiments have been performed on 312 different patient cases selected from the cancer genome atlas (TCGA) and have achieved superior performances over state-of-the-art texture descriptors and baseline methods including deep learning models for prostate cancer Gleason grading.

Index Terms—Prostate cancer, Medical image analysis, Texture features, Image classification.

1 Introduction

PROSTATE cancer is the second most common cancer in men and the fourth most common tumor type worldwide [1]. Although there have been significant changes in clinical and histologic diagnosis of prostate cancer, the Gleason grading of tumor biopsies remains one of the most powerful prognostic predictors in prostate cancer [2]. Since prostate cancer is a biologically heterogeneous disease with variable molecular alterations, the manual grading by pathologists often suffers from inter- and intra-observer variations. Therefore, the ability to automatically assign Gleason scores from diagnostic pathology slides would have significant impacts on clinical decision making, and prediction of patient outcomes.

The Gleason grading system defines five histological patterns from least aggressive (i.e., grade 1) to most aggressive (i.e., grade 5), based on gland structures in the tumor biopsy [3]. Since most tumors typically have two patterns, the original Gleason score is assigned by adding the two most common patterns in a tumor, with scores ranging from 2 to 10 [2]. However, the current application of Gleason grading system has been changed from the original version. Nowadays pathologists mainly assign Gleason scores 6-10 based on the two dominant tumor patterns, since assignment of Gleason scores 2-5 has poor reproducibility and poor correlation with radical prostatectomy grade [2]. Gleason score along with other clinical variables are usually used to create risk stratification for prostate cancer patient management. For example, patients with Gleason score 6 or below is typically considered as low risk. Patients with Gleason score 7 is considered as intermediate risk, and patients with Gleason score 8 or above is considered as high risk. Fig. 1 shows examples of tumor regions with different Gleason scores. As observed in Fig. 1, the textural appearance for different grade of prostate cancers varies from each other due to abnormal changes of gland structures. Based on gland and tumor patterns, there have been a plethora of published studies addressing automatic prostate cancer grading problem. The existing studies can be broadly divided into three main categories: gland-nuclei-based, deep learning and texture-based approaches.

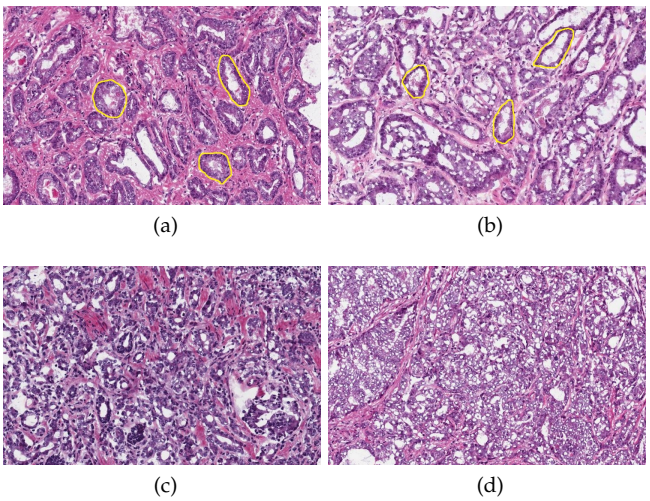

Fig. 1. Example of Gleason patterns. (a) Gleason score 6 (low risk). (b) Gleason score 7 (intermediate risk). (c) Gleason score 8 (high risk). (d) Gleason score 9 (high risk). Note that in (a)(b) the superimposed yellow contours highlight a few manually labeled glands.

Gland-nuclei-based techniques attempt to determine Gleason scores by computing shape and structural information of segmented glands and nuclei in the image. For example, Nguyen et al. [3] proposed a method that first

• *H. Xu, S. Park and T. H. Hwang were with the Department of Quantitative Health Sciences, Cleveland Clinic, Cleveland, OH, 44195. E-mails: xuh3@ccf.org, parks@ccf.org, and hwangt@ccf.org*

Manuscript received xx xx, xxxx; revised xx xx, xxxx.

segments glands by incorporating nuclei distribution into a normalized graph cut framework and then assigns Gleason scores based on structural and elliptical features of segmented glands. They reported about 87% accuracies for grade 3 and 4 classifications. Niazi et al. [4] proposed a set of visually meaningful features to differentiate between low (≤ 6) and high (≥ 8) Gleason scores of prostate cancer. These features include such as the shortest path from epithelial nuclei to the closest luminal and the ratio of epithelial nuclei over the total number of nuclei in the image. They reported a high classification accuracy (above 90%), but the most challenging cases of Gleason score 7 were not included for evaluation. These gland-nuclei-based techniques usually fail to process and identify high Gleason scores, since tumor regions are merged together and there are no clear glands for detection in high Gleason patterns (see Fig. 1(d)).

Deep learning models attempt to predict prostate cancer Gleason scores by utilizing image features extracted from deep convolutional networks. Källén et al. [5] presented a technique that first extracts image features by transfer learning on OverFeat model and then predicts prostate cancer Gleason grades by using random forest and SVM classifiers. Jimenez-del-Toro et al. [6] trained the GoogLeNet model on 141 TCGA whole slide images, and achieved about 78% classification accuracies in distinguishing Gleason scores $\geq$8 and $\leq$7 on 46 testing images. Arvaniti et al. [7] applied a transfer learning approach for automated Gleason grading of H&E stained prostate cancer tissue microarrays. Although deep learning models are very powerful to learn image features, they generally require a large number of well-annotated training samples which are difficult to obtain, especially for pathology slides [8]. For instance, TCGA pathology slide only has patient label, i.e., it does not have ground truth labels for small tissue regions or patches that are usually required for training a deep learning model.

Texture-based techniques attempt to determine Gleason scores by capturing the textural difference caused by gland changes in different grade of tumor biopsy slides. The widely-used textural features are computed from gray level co-occurrence matrix, multi-wavelet transform, fractal analysis and texton maps. Huang et al. [9] proposed to classify prostate pathological image with different Gleason grades by applying fractal analysis to describe histological variations within the image. Khurd et al. [10] proposed a texture classification technique that characterizes different Gleason grades by clustering extracted filter responses (at each pixel) into textons. The studies [9], [10] only analyzed manually selected regions of interest rather than whole pathology slides. The feature representation from selected image patches is likely to bring sampling bias to prostate cancer grading. However, texture-based techniques have the advantages that they are capable of describing high Gleason grade patterns due to encoding textural appearance from pixel level rather than object level. Therefore, we design a group of textural features termed as completed and statistical local binary pattern (CSLBP) descriptors, which are shown to be more effective in encoding image texture patterns for prostate cancer Gleason score predictions.

In this paper, we present an automated technique for prostate cancer Gleason score classification from digitized tumor tissue pathology slides. There are two main contributions from this work:

- To the best of our knowledge, this is the first study to perform Gleason grading of prostate cancer patients using computerized textural features from whole histopathology slide images.
- A set of completed and statistical local binary pattern (CSLBP) descriptors are proposed for textural analysis in prostate pathology images, which are shown to be superior to other traditional textural features for Gleason grading. These CSLBP descriptors are general and applicable to other medical image analysis problems.

The rest of this paper is organized as follows. Section 2 describes the presented framework for Gleason score prediction. Section 3 provides experimental results, followed by conclusion in Section 4.

2 Proposed Method

Fig. 2 describes an overview of our proposed technique. As observed in Fig. 2, the presented technique mainly consists of three modules: image preprocessing, feature extraction and image classification. (1) Since the WSI has a huge size and is usually heterogenous with different Gleason patterns, the WSI is preprocessed and divided into a set of non-overlapping image blocks. (2) The texture features encoded by completed and statistical local binary pattern (CSLBP) descriptors are computed from selected image blocks of the WSI. The CSLBP descriptors encode local image patterns by analyzing the difference between every central pixel and pixels of its surrounding neighborhoods, which is more robust to encode image textures across different color appearance and quality images. (3) Since the small image blocks do not have ground truth labels (i.e., Gleason scores), image features computed across different image blocks of the WSI are integrated together. The SVM classifier is then trained on patient level, which assigns patient slides to different Gleason scores (i.e., low, intermediate or high risk). In the following subsections, we provide details of our proposed technique.

2.1 Image preprocessing

A whole pathology slide with a high resolution typically has a large volume size (e.g., about 1GB for 20$\times$), which makes processing it directly with computerized algorithms a challenge [11]. In this module, the pathology image with 2.5$\times$ resolution is selected using the openslide library [12] for efficient processing. The whole pathology slide is then divided into a set of non-overlapping image blocks for subsequent processing. The three steps of this module are as follows.

2.1.1 Stain separation

Given an RGB color pathology image, the color deconvolution method is first applied to separate the image into hematoxylin (H) and eosin (E) channels, respectively [13]. Since most of biological information (e.g., nuclei) is included in H channel, the H channel is then converted into a gray scale image, H_g, for subsequent processing. Fig. 3(a) shows a H&E stained prostate pathology image, and Fig. 3(b) shows the gray scale image H_g.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

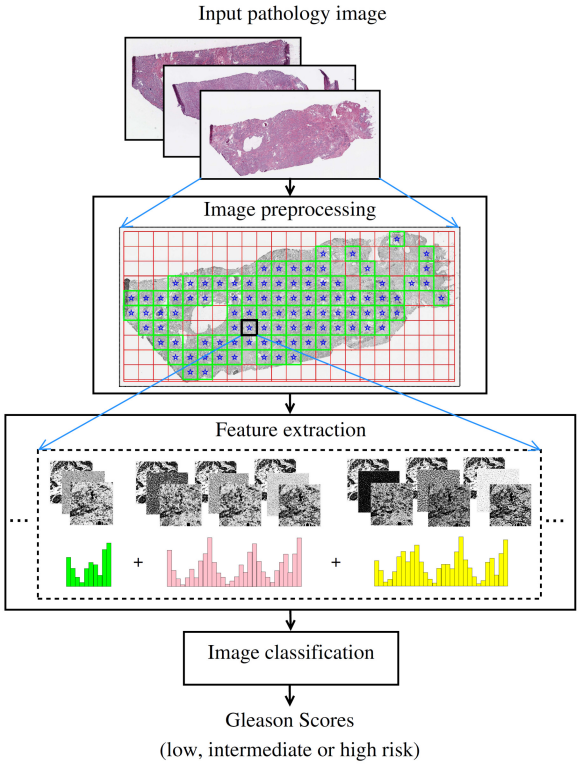

Fig. 2. Pipeline of the proposed technique. The technique has three main modules: image preprocessing, feature extraction and image classification.

2.1.2 Multi-thresholding

It is observed from Fig. 3(b) that image H_g mainly includes three classes of pixels: background (white), tissue stroma (gray) and nuclei (dark). Based on this observation, the image H_g is segmented into three levels by two thresholds, $0 < \tau_1 < \tau_2 < 255$. The thresholds τ_1 and τ_2 are automatically determined using Otsu's method [14], which computes the optimal thresholds by maximizing the inter-class variances. The pixels with intensities below the threshold τ_1 are determined as nuclei pixels, while the pixels with intensities between two thresholds τ_1 and τ_2 are determined as stroma pixels. In Fig. 3(c), the white pixels represent segmented image background pixels, whereas green and blue pixels represent segmented stroma and nuclei pixels, respectively.

2.1.3 Image tiling

The bounding box [13] of the prostate tissue region (i.e., stroma and nuclei regions) is computed based on image thresholding. In Fig. 3(c) the red rectangle is the determined bounding box for prostate tissue. The image region within the bounding box is then divided into a number of non-overlapping blocks. Let us assume that there are K image blocks, and each block has a size of $u \times v$ pixels. The ith image block is selected for subsequent feature analysis only if it satisfies the following two inequalities:

$$\kappa^i_{tis} > T_1 \quad (1)$$

$$\kappa^i_{nuc} > T_2 \quad (2)$$

where $\kappa^i_{tis} = N^i_{tis}/uv$ and $\kappa^i_{nuc} = N^i_{nuc}/uv$. N^i_{tis} and N^i_{nuc} represent the number of prostate tissue and nuclei pixels within the ith image block, respectively. T_1, T_2 are two thresholds, which were empirically set as $T_1 = 0.9$ and $T_2 = \max\{\kappa^i_{nuc}\} * 0.25$, $1 \leq i \leq K$, in this work. The above two inequalities are applied such that image blocks containing too many background regions (i.e., >10%) or few nuclei pixels are discarded for further feature analysis, as these image blocks tend to include non-tumor regions. The threshold T_2 is adaptively determined as a quarter of the largest nuclei density among all blocks in the image. Note that nuclei densities vary greatly among different patient pathology slides and tumor stages, and hence an adaptive threshold T_2 helps in selecting enough tiles for feature analysis across images with diverse tumor nuclei densities. In Fig. 3(d) red and blue squares highlight divided image blocks, where $u = v = 128$. Blue squares in Fig. 3(d) indicate selected image blocks that satisfy the above two inequalities (1)(2). As observed in Fig. 3(d), the blocks mainly belonging to tumor regions are selected for further texture analysis, while the blocks containing too many background or stroma regions have been discarded. Let selected image blocks for further analysis be denoted by B_i, $1 \leq i \leq S$, where S is the number of selected image blocks.

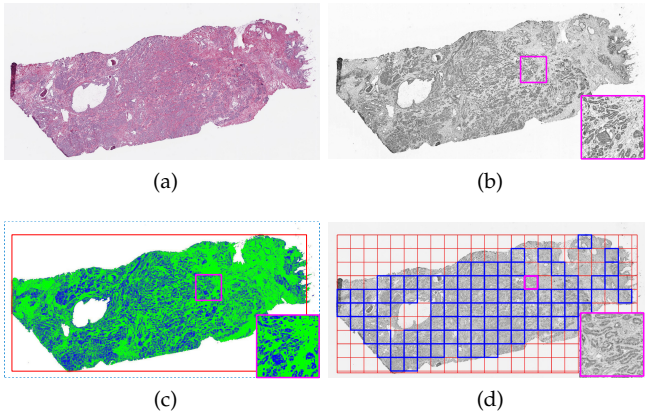

Fig. 3. Illustration of image preprocessing. (a) Prostate pathology slide. (b) Gray scale image H_g. (c) Image segmentations, where white pixels represent image background, green and blue pixels represent stroma and nuclei regions. (d) Image tiling results. Note that in (d) image blocks highlighted by blue squares are selected for subsequent texture analysis.

2.2 Feature extraction

After obtaining image blocks from the WSI, texture analysis is performed to capture different Gleason patterns. In this module, tumor blocks B_i, $1 \leq i \leq S$ with 5.0× magnification are determined based on the openslide library [12] to perform feature extraction. First a set of texture features is computed from every selected image block using our proposed completed and statistical local binary pattern (CSLBP) descriptors. The computed texture features from all blocks of the WSI are then integrated together to characterize patient tumor severity. The details of this module are described below.

2.2.1 CSLBP

The presented CSLBP is extended from completed local binary pattern (CLBP) [15], and includes one more subcate-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

gory termed as statistical local binary pattern (SLBP). In the following, we first briefly describe the CLBP and then detail the SLBP.

CLBP: The CLBP is the completed modeling of local binary pattern (LBP) operator [16], which encodes textural information by computing sign and magnitude differences between the central pixel and its surrounding neighbors. Fig. 4 illustrates the computation of CLBP. As observed in Fig. 4, given a center pixel g_c and its p circularly (with radius r) and evenly spaced neighbors g_n, $n = 0, 1, ..., p-1$, the CLBP encoding consists of three components:

$$CLBP\_C\left(g_c\right) = s\left(g_c - t\right) \tag{3}$$

$$CLBP\_S\left(g_c\right) = \sum_{n=0}^{p-1} s\left(g_n - g_c\right) 2^n \tag{4}$$

$$CLBP\_M\left(g_c\right) = \sum_{n=0}^{p-1} s\left(m_n - c\right) 2^n \tag{5}$$

where $CLBP\_C\left(g_c\right)$, $CLBP\_S\left(g_c\right)$ and $CLBP\_M\left(g_c\right)$ encode the center, sign and magnitude information for pixel g_c, respectively. $s\left(\cdot\right)$ is the sign function, i.e.,

$$s\left(x - y\right) = \begin{cases} 1, & x \geq y \\ 0, & x < y \end{cases} \tag{6}$$

The threshold t is the average gray intensity of the whole image block (at 5× magnification in this study). m_n is the absolute difference between the center pixel and its neighbor g_n, and c is the mean value of m_n over the whole image block [15].

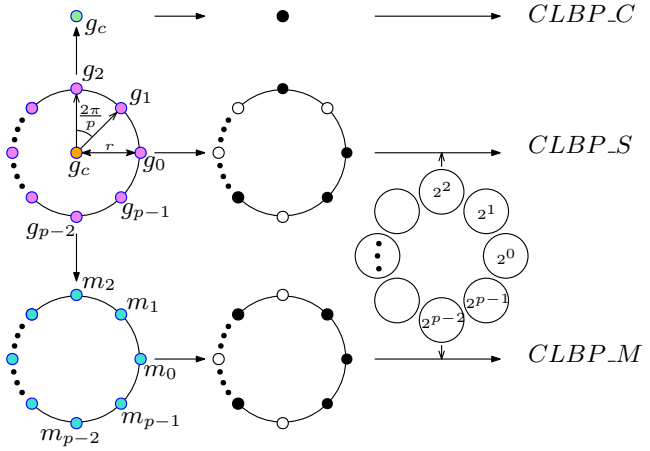

Fig. 4. Illustration of completed local binary pattern (CLBP).

After computing $CLBP\_S$ and $CLBP\_M$ for each pixel in image block B_i, the rotation invariant uniform $riu2$ encoding scheme [16] is applied, which reduces the values of $CLBP\_S$ and $CLBP\_M$ from the range $0 \sim 2^p - 1$ to $0 \sim p + 1$. For more details about $riu2$ encoding scheme, please refer to references [16]–[18]. In Fig. 7, the top row shows an example of gray scale image block B_i, while the second row shows three feature maps $CLBP\_C$, $CLBP\_S$ and $CLBP\_M$, respectively. To ensure a relatively small feature dimension, a joint 2D histogram $CLBP\_M/C$ is first built from feature maps $CLBP\_C$ and $CLBP\_M$. The 2D histogram $CLBP\_M/C$ is then converted to a 1D histogram and concatenated with the histogram of feature map $CLBP\_S$ to generate a joint histogram, denoted by $CLBP\_S\_M/C$. In total, $(p + 2) * 3$ histogram features are extracted from CLBP descriptors.

SLBP: Although CLBP is capable of capturing microstructure information (e.g., corner and edge) in the image, it is susceptible to image noise and it is difficult to capture macrostructure information due to a small number of neighboring pixels analyzed. To alleviate these limitations, we propose the SLBP to encode neighboring pixels in a large surrounding region. Formally, given the center pixel g_c and its p surrounding neighborhoods G_n, $n = 0, 1, ..., p-1$, the $SLBP\_C\left(g_c\right)$, $SLBP\_S\left(g_c\right)$, and $SLBP\_M\left(g_c\right)$ descriptors are computed as follows:

$$SLBP\_C\left(g_c\right) = s\left(g_c - t\right) \tag{7}$$

$$SLBP\_S\left(g_c\right) = \sum_{n=0}^{p-1} s\left(f\left(G_n\right) - g_c\right) 2^n \tag{8}$$

$$SLBP\_M\left(g_c\right) = \sum_{n=0}^{p-1} s\left(f\left(M_n\right) - \varphi(\mathrm{c})\right) 2^n \tag{9}$$

where $SLBP\_C\left(g_c\right)$ encodes the center pixel g_c which is the same as the CLBP shown in Eq. (3). $SLBP\_S\left(g_c\right)$ and $SLBP\_M\left(g_c\right)$ encode the sign and magnitude information of g_c by analyzing its surrounding neighborhoods. G_n represents a set of pixels in the nth neighborhood of g_c, and M_n represents a set of absolute differences between pixels in G_n and g_c. $f\left(\cdot\right)$ represents a statistical filter function. $\varphi(\mathrm{c})$ is the threshold and computed as:

$$\varphi(\mathrm{c}) = \frac{1}{4puv} \sum_{c=1}^{4uv} \sum_{n=0}^{p-1} \left| f\left(G_n\right) - g_c \right| \tag{10}$$

where u and v are the image block size determined during image tiling at 2.5× magnification (see Section 2.1.3). Since feature extraction is performed at 5.0× magnification, each image block B_i has $4uv$ pixels.

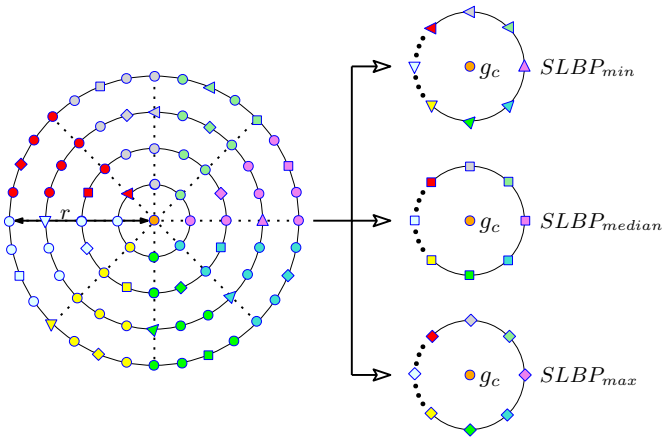

Fig. 5. Illustration of statistical local binary pattern with orientational neighborhood (O-SLBP).

Note that different kinds of neighborhood G_n could be explored to analyze surrounding patterns for pixel g_c. In this work, we propose two surrounding neighborhoods: orientational neighborhood and radial neighborhood. Fig. 5

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

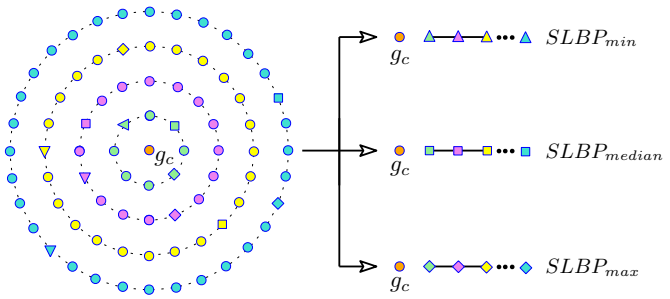

Fig. 6. Illustration of statistical local binary pattern with radial neighborhood (R-SLBP).

illustrates SLBP with orientational neighborhood (henceforth referred to as O-SLBP), where the surrounding region of pixel g_c is evenly divided into 8 (i.e., $p = 8$) orientational neighborhoods represented by different color of circles, triangles, squares or diamonds. Fig. 6 illustrates SLBP with radial neighborhood (henceforth referred to as R-SLBP), where the surrounding region of pixel g_c is divided into 4 (i.e., $p = 4$) radial neighborhoods based on the distance between every surrounding pixel and the center pixel g_c. As observed in Figs. 5 and 6, the number of pixels within each neighborhood G_n depends on radius r of the surrounding region and the type of neighborhood applied.

Similarly, different types of statistical function $f(\cdot)$ could be utilized to filter out pixels from the surrounding neighborhood G_n. In this work, we propose to use three filter functions: min, $median$ and max, to select pixels from neighborhoods G_n, $0 \leq n \leq p-1$. In Figs. 5 and 6, the triangles, squares and diamonds represent the minimum, median and maximum pixels in each neighborhood, respectively. As observed in Figs. 5 and 6, after selecting the minimum, median and maximum pixels from neighborhoods, three neighboring patterns of the center pixel g_c are generated: $SLBP_{min}$, $SLBP_{median}$ and $SLBP_{max}$. These three neighboring patterns are separately encoded with Eqs. (7)(8)(9). For the O-SLBP shown in Fig. 5, the $riu2$ encoding scheme [16] is applied on $SLBP\_S$ and $SLBP\_M$, since neighbors circularly surround the center pixel g_c, which is the same surrounding pattern as CLBP operator. Nine feature maps corresponding to $SLBP_{min}$, $SLBP_{median}$ and $SLBP_{max}$ of orientational neighborhoods are shown in the third row of Fig. 7. After obtaining these feature maps, the joint histograms $SLBP_{min}\_S\_M/C$, $SLBP_{median}\_S\_M/C$, and $SLBP_{max}\_S\_M/C$ are separately computed and concatenated together, which results in $(p+2)*3*3$ dimensional histogram features together. For the R-SLBP shown in Fig. 6, since statistical neighbors are selected along radial directions and they are invariant regrading image rotations, it is not necessary to further apply the $riu2$ encoding scheme on $SLBP\_S$ and $SLBP\_M$. Nine feature maps corresponding to $SLBP_{min}$, $SLBP_{median}$ and $SLBP_{max}$ of radial neighborhoods are shown in the fourth row of Fig. 7. After obtaining these feature maps, joint histograms $SLBP_{min}\_S\_M/C$, $SLBP_{median}\_S\_M/C$, $SLBP_{max}\_S\_M/C$ are separately computed and concatenated together, which results in $2^p * 3 * 3$ dimensional histogram features together.

2.2.2 feature integration

After computing CSLBP texture features from all image blocks B_i, $1 \leq i \leq S$, statistical distribution measures are computed for every CSLBP feature such that feature values across different image blocks of the WSI are integrated together. These statistical measures include mean, standard deviation, skewness and kurtosis [19]. Skewness is a measure of the asymmetry of the data around the sample mean. Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. Consequently, after computing 4 statistical measures across S image blocks for all CSLBP features, $4 * \left[(p_c+2)*3+(p_s^o+2)*9+2^{p_s^r}*9\right]$ dimensional feature vector per patient pathology slide is obtained, where p_c, p_s^o and p_s^r represent the number of surrounding neighbors for CLBP, O-SLBP and R-SLBP, respectively. Let $\mathrm{X} = (x_1, x_2, \ldots, x_N)$ denote the finally obtained feature vector for a patient pathology slide, where N indicates the number of feature components.

2.3 Image classification

With the computed textural features $\mathrm{X} = (x_1, x_2, \ldots, x_N)$ from a WSI, the patient tumor is now ready to be classified into different categories of Gleason scores. In this work, principal component analysis (PCA) is first applied to reduce feature dimensions [19], i.e.,

$$\mathrm{X}^* = \mathrm{X}W_L \tag{11}$$

where $\mathrm{X}^* = (x_1^*, x_2^*, \ldots, x_L^*)$ is the PCA transformed feature vector. W_L is the PCA transformation matrix computed from the training dataset, and L indicates the number of principal components selected by PCA transformation. The PCA is applied for two main reasons. (1) high dimensional features are reduced into low dimensional space in order to prevent over-fitting to the training dataset. (2) PCA helps to reduce feature noise and remove redundant features (i.e., not contributing much to the discrimination power). After feature dimension reduction, feature standardization is performed on each feature to make its values have zero mean and unit variance. Finally, the "one-against-one" multi-class support vector machine (SVM) method [13] is used to perform prostate cancer patient classification. Let k denote the number of classes for classification. In the multi-class SVM method, we construct $k(k-1)/2$ SVM classifiers, and each classifier is trained on data from two classes during the training phase. After obtaining $k(k-1)/2$ SVM classifiers, the test patient image will be labeled by all of them in the testing phase. The test patient is predicted to be in the class that is labeled by most of the binary SVM classifiers.

3 Experiments and Evaluations

In this section, we perform experiments by evaluating three-class distinctions (e.g., high, intermediate, or low risk patients) about prostate cancer Gleason score predictions. To demonstrate the efficacy of our proposed CSLBP descriptors, we compare our technique with several commonly-used texture descriptors and baseline deep learning models. In addition, we evaluate the proposed texture descriptor with different classifiers to test its robustness across different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

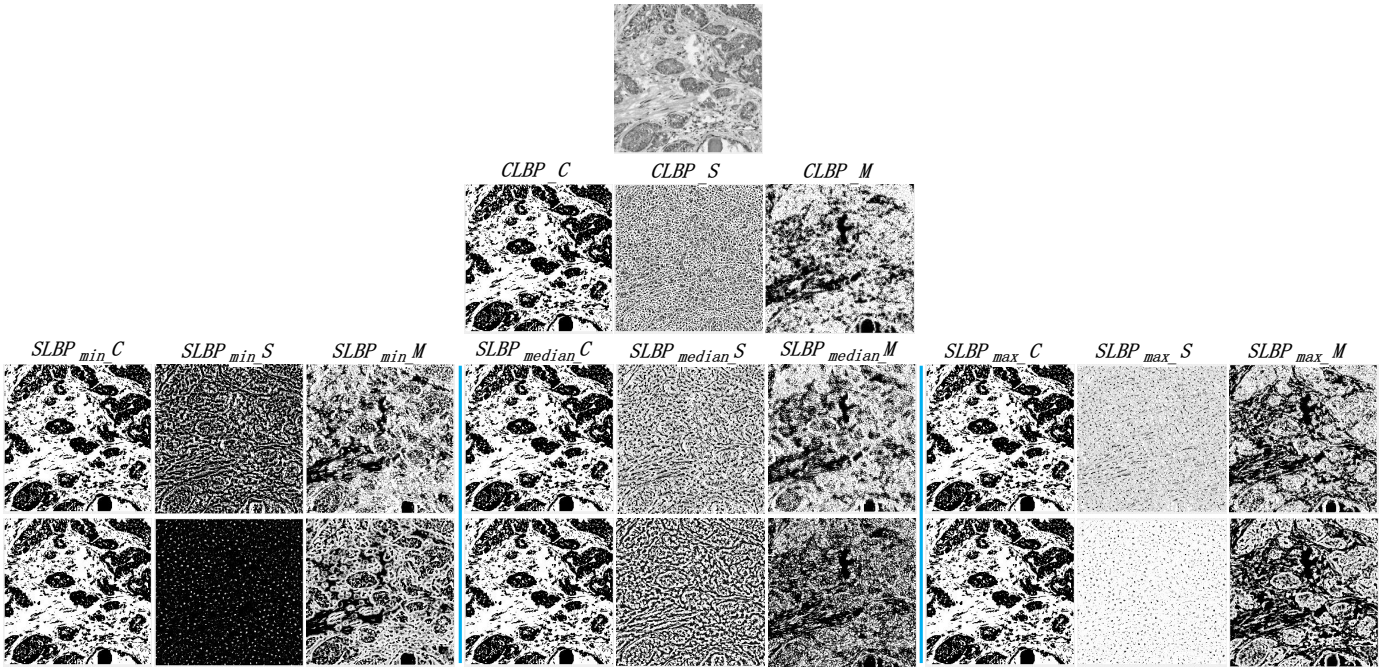

Fig. 7. Example of feature maps obtained from CSLBP descriptor. First row: gray input image block; second row: CLBP feature maps; third row: O-SLBP feature maps; fourth row: R-SLBP feature maps.

classification settings. Details of this module are described below.

3.1 Image dataset

All whole slide prostate pathology images were obtained from the cancer genome atlas (TCGA) [1], which were stained with hematoxylin and eosin (H&E). TCGA prostate data was derived from multiple institutions over many years, and includes a cohort of 500 prostate adenocarcinoma patients in total. However, some TCGA pathology slides have relatively poor qualities due to tissue distortions and image artifacts such as tissue folders and pen marks [20], which are not suitable for computerized pathological image analysis. Based on image preprocessing and visual examinations, 312 acceptable patient pathology slides are used in this study. The ground truths of Gleason scores are also provided by TCGA on patient records. Among 312 patient slides, 32 patients are diagnosed as low risk with Gleason score 6, 141 patients are diagnosed as intermediate risk with Gleason score 7 and 139 patients are diagnosed as high risk with Gleason score $\geq$ 8. Since the number of low risk patients (i.e., with Gleason score 6) is far less than that of intermediate or high risk patients, we augment the number of pathology slides for low risk patients to make three groups of images roughly balanced. Specifically, after computing the bounding box of prostate tissue pixels for every patient pathology slide (see the red rectangle shown in Fig. 3(c)), the bounding box is shifted towards bottom-right direction by 30, 60 and 90 pixels, respectively. The image tiling in Section 2.1.3 is then separately performed based on shifted bounding boxes. Thus 1 patient pathology slide is augmented to 4 images, and low risk patient slides are increased to 128 after augmentation. Note that the bounding box shifting has altered the selection of tumor tiles from the same patient pathology slide.

3.2 Parameter settings

There are two key parameters for CSLBP texture descriptors: radius r and number of neighbors p. Table 1 illustrates parameter settings for (r, p) values of CLBP and SLBP descriptors. Note that large (r, p) values would result in high computational complexity and feature dimensions, while (r, p) values that are too small may fail to capture macrostructure information in the image. These (r, p) values are determined by balancing feature representations and computational complexities. As observed in Table 1, 288 CSLBP textural features are computed from every selected image block of the WSI. After computing 4 statistical measures across all selected image blocks (see Section 2.2.2), one patient pathology slide corresponds to 1152 dimensional features. Since the number of patient samples is much lower than that of feature dimensions, the PCA is applied for feature dimension reduction to against over-fitting. The number of feature dimensions selected by PCA is experimentally determined based on the performance (see Section 3.6).

The SVM classification is implemented using Matlab Statistics and Machine Learning toolbox [21], where two types of kernels: Polynomial and Gaussian kernels, are explored in this study. For both Polynomial and Gaussian kernels, the kernel scale k_s is a key parameter that may affect SVM classification performance. To select a suitable k_s value, a heuristic procedure [22], [23] provided by Matlab toolbox is applied, which automatically determines k_s values based on the training set during cross validations. Except kernel scale k_s, all other parameters are set as default values of Matlab toolbox, since those parameter values are usually working well on different classification prob-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

lems [13]. The preliminary version of Matlab programs for this study and comparisons are publicly accessible via the website: https://github.com/hwanglab/tcga-prad-cslbp/.

TABLE 1
Parameter settings for texture analysis

Descriptors	Radius r	Neighbors p	Feature dimensions
CLBP	2	16	54
O-SLBP	4	8	90
R-SLBP	4	4	144

3.3 Baseline texture descriptors and deep learning models for Gleason grading prediction

To evaluate the efficacy of proposed CSLBP descriptor for prostate cancer Gleason grading, the proposed technique is compared to several widely-used texture descriptors for medical image analysis and deep learning methods, which are described below:

(1) Fractal Analysis based technique [9] (henceforth referred to as FA technique). For the FA technique, four groups of grid size including $\{2, 4, 8\}$, $\{8, 16, 32\}$, $\{32, 64, 128\}$ and $\{2, 4, 8, 16, 32, 64, 128\}$ are used for fractal analysis by the differential box-counting method [24]. 4 fractal dimension texture features are derived from image intensity difference and image entropy, respectively, which results in 8 texture features for each selected image block. The mean, standard deviation, skewness and kurtosis of each feature are then computed across all image blocks of the WSI, therefore each whole patient slide corresponds to 32 dimensional features. These 32 features are used by the SVM classifier for prostate cancer grading. The parameters of the SVM classifier are tuned and set following the procedure of our proposed technique.

(2) Histogram, Haralick and Gabor filter [25], [26] based technique (henceforth referred to as HHG technique). In our implementation, we first compute 6 first-order histogram features, 60 second-order Haralick features and 12 Gabor filter related texture features from every selected image block. The mean, standard deviation, skewness and kurtosis of each feature are then computed across all image blocks of the WSI, which results in 312 dimensional features for each patient slide. During evaluation, we test to directly apply these 312 dimensional features with the SVM classifier. In addition, we test to use PCA for feature dimension reduction and then apply the SVM classifier for prostate cancer Gleason grading. Likewise, the parameters of the SVM classifier are tuned and set following the procedure of our proposed technique.

(3) VGG16 [27] based Transfer Learning. **We test two strategies of transfer learning on VGG16 model. Fig. 8 shows the architecture of our first transfer learning model (henceforth referred to as VGG16-TL1), while Fig. 9 illustrates our second transfer learning model (henceforth referred to as VGG16-TL2).**

For the VGG16-TL1, we reuse all convolutional and pooling layers from the original VGG16 model. Meanwhile, we add a 1×1 convolutional layer, 1 dropout layer and 2 fully connected layers. **The 1×1 convolutional layer is added for reducing the number of neurons in the fully connection layer, while the dropout layer is added for suppressing overfitting [28]. The fully connection layers with ReLU activation functions are added for nonlinear transformation of extracted image features.** During training, the trainable parameters in the original VGG16 model are frozen, while only the parameters in our added layers are fine-tuned. To train the VGG16-TL1, the WSI is divided into many small image tiles which are assigned with the same label as the whole patient slide. The training process is performed by using stochastic gradient descent algorithm (with a learning rate 0.001) by minimizing the categorical cross entropy loss function. The batch size is set as 64. The training is performed by 100 epochs with early stopping if validation performance is not improved. During testing, each WSI has many small tiles and each tile has a predicted probability. The final prediction for a patient slide is obtained by averaging predictions on small tiles of the WSI.

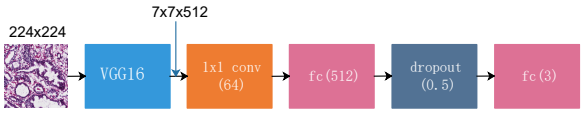

Fig. 8. Illustration of VGG16-TL1 model.

For the VGG16-TL2 (see Fig. 9), we fine-tune trainable parameters in last three 3×3 convolutional layers of the VGG16 model and our added layers on top of VGG16 model together. The shallow convolutional layers in VGG16 generally provide low-level features that could be used across different imaging datasets. The deep convolutional layers (i.e., the last three convolutional layers) are fine-tuned such that the learned high level features could adapt to our dataset. Due to the large amount of trainable parameters, image augmentations including rotation, zooming, flipping and color based augmentations [29] are randomly applied with training VGG16-TL2. All other training and testing settings follow the same procedure as the VGG16-TL1.

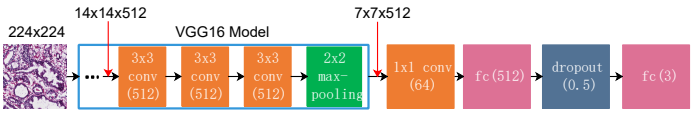

Fig. 9. Illustration of VGG16-TL2 model.

(4) VGG16 based Deep Learning (henceforth referred to as VGG16-DL). Fig. 10. shows the architecture of our trained VGG16-DL model. As shown in Fig. 10, VGG16-DL has a similar architecture with the original VGG16 model [27], except that the number of feature map channels in convolutional layers and the number of neurons in dense layers are reduced. As with the VGG16-TL1, the WSI is divided into many small tiles to train the VGG16-DL. Because there is a large number of trainable parameters, image augmentations as used for training the VGG16-TL2 are randomly applied for training the VGG16-DL. All other training and testing settings follow the same procedure as the VGG16-TL1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

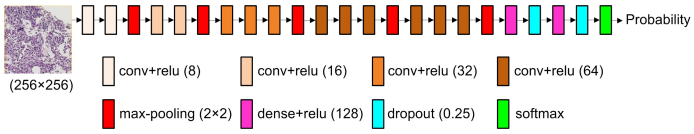

Fig. 10. Architecture of VGG16-DL model.

3.4 Evaluation metrics

In this study, prostate cancer image samples are divided into 3 risk groups: low risk (Gleason score 6), intermediate risk (Gleason score 7) and high risk (Gleason score 8 or above) [2]. During evaluation, if the automatic grading is the same as the expert class label, it is considered as a correct classification. The performance of the automatic technique for Gleason score prediction is evaluated by using classification accuracy ACC, which is defined as:

$$ACC = \frac{\sum_s \sum_{j=1}^{N^s} 1\left(B_j^s = s\right)}{\sum_s N^s} \times 100\% \tag{12}$$

where s indicates a class of one of three risk groups, i.e., $s \in \{LOW, ITM, HIGH\}$. N^s is the number of samples belonging to ground truth class s, B_j^s represents the automatically classified result for the jth sample in the class s and $1\left(\cdot\right)$ is the indicator function, i.e.,

$$1\left(x\right) = \begin{cases} 1 & if\, x\, is\, true \\ 0 & otherwise \end{cases} \tag{13}$$

We also compute the area under the receiver operating characteristic (ROC) curve to evaluate the performance of computerized techniques. For the three-class classification problem, three binary classifications are created for generating ROC curves. Specifically, each time we consider one class of patient slides as positive samples and the remaining two classes as negative samples to generate a ROC curve. This process is repeated three times and thus three ROC curves are obtained. Let AUC_k denote the area under the kth ROC curve, where $1 \leq k \leq 3$. The mean AUC for the three ROC curves is computed as:

$$AUC = \frac{\sum_{k=1}^{3} AUC_k}{3} \tag{14}$$

By using the above two criteria, we perform the 3-fold cross validation and repeat it 50 times for evaluation of the FA, HHG and proposed techniques. The average values of different evaluation criteria are used as final results. Due to the high computational time, 3-fold cross validation is only run 1 time for evaluation of VGG16-TL1, VGG16-TL2 and VGG16-DL deep learning models.

3.5 Classification results

Table 2 lists ACC and AUC values for evaluation of three-class Gleason score predictions by different methods. In Table 2, the numerical values within parenthesis are obtained by using SVM classifier with polynomial kernels. The numerical values correspondingly outside parenthesis are obtained by using SVM classifier with Gaussian kernels. The HHG+PCA and proposed techniques apply PCA for feature dimension reduction, where 50 feature components are selected based on experimental performances. As observed in Table 2, the FA technique provides the poorest performance, with a ACC value around 61% and a AUC value around 0.8. The HHG technique either with or without PCA for feature dimension reduction provides intermediate performances, with AUC values around 0.85. The VGG16-TL1 and VGG16-DL provide marginally poor performances than the HHG+PCA technique, with the ACC values around 65% and the AUC values below 0.84. **The VGG16-TL2 provides the best performance among three deep learning baseline methods, with about 70% accuracy and 0.85 AUC value.** In our experiments, there are 12,115 image blocks belonging to low risk patients, 17,810 image blocks belonging to intermediate risk patients and 19,908 image blocks belonging to high risk patients. Due to unbalanced datasets, deep learning predictions tends to bias towards high risk patients (see Fig. 11). In addition, the patient slide is usually heterogenous with different Gleason score patterns. To train deep learning models, all image blocks have to be assumed with the same label as the patient slide, which is not always true [30]. Compared with existing techniques and baseline deep learning models, the proposed technique provides marked improvement, which achieves about 77% accuracy and 0.93 of AUC value when SVM classifier with Gaussian kernel is applied. There are two main aspects that make the proposed technique provide a good performance. (1) Compared with traditional texture representation such as histograms or Haralick features, the CSLBP descriptors are more robust in terms of color variations and image noise that often exist in TCGA pathology images. (2) Unlike deep learning models that are trained on patch level, our SVM classifier is trained on patient slide level, which alleviates the limitation that small tiles are not guaranteed to have the same label as the patient slide.

TABLE 2
Comparison of three-class prediction performance including augmented patient slides

Techniques	ACC (%)	AUC
FA [9]	61.11 (61.15)	0.802 (0.783)
HHG [25]	63.51 (68.05)	0.839 (0.863)
HHG+PCA	65.10 (67.10)	0.860 (0.857)
VGG16-TL1	64.46	0.807
VGG16-TL2	69.12	0.852
VGG16-DL	65.44	0.837
Proposed	77.12 (76.48)	0.932 (0.920)

Fig. 11 shows ROC curves obtained by different techniques for three-class distinctions that include predictions on augmented patient slides. In Fig. 11, performances obtained by SVM classifiers with Gaussian kernels are provided. In Fig. 11(a) the low risk group is considered as the positive class, while the intermediate and high risk groups are considered as the negative class. Similarly, in Figs. 11(b) and (c) the intermediate and high risk groups are separately considered as positive class, while the remaining two risk groups are considered as negative class. As observed in Fig. 11, the proposed technique provides an overall remarkably better performance than existing techniques and three implemented baseline deep learning models. It is noted in Fig. 11(c) that VGG16-TL1, VGG16-TL2 and VGG16-DL

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

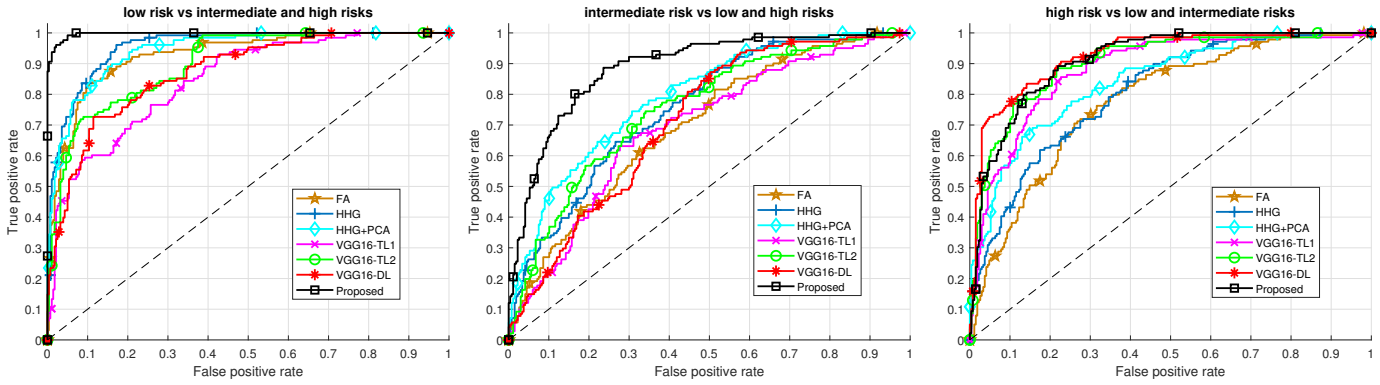

Fig. 11. ROC curves of different techniques with predictions including augmented patient slides. (a) Low risk vs intermediate and high risk groups. (b) Intermediate risk vs low and high risk groups. (c) High risk vs low and intermediate risk groups. SVM classifiers used by FA, HHG, HHG+PCA and proposed techniques are set with Gaussian kernels in these figures.

achieve good performances in predicting high risk patients mainly due to the high proportion of image blocks belonging to the high risk patient group during training deep learning models.

Due to the small number of low risk patient slides (i.e., 32 patients), three-class distinctions in Table 2 are performed by augmenting on low risk patient slides to avoid severe unbalanced issues. To further evaluate the performance of different techniques, we perform experiments by training the model with augmentations but testing the model on patients without augmentations. Table 3 lists comparison results of different techniques that exclude predictions on augmented patient slides. As with the Table 2, the numerical values inside and outside the parenthesis in Table 3 are obtained by using SVM classifiers with polynomial and Gaussian kernels, respectively. For the HHG+PCA and proposed techniques, 50 feature components are selected based on PCA transformation. As observed in Table 3, the FA technique provides the poorest performance, with a ACC value around 54% and a AUC value around 0.75. The HHG either with or without PCA provides better performances than the FA technique. Particularly, the HHG with PCA for feature dimension reduction achieves around 61.5% accuracy and 0.82 AUC value, which is better than using all 312 HHG texture features. The VGG16-TL1 and VGG16-DL provide similar accuracy (around 68%), but the VGG16-DL provides a higher AUC value (0.846) than the VGG16-TL1. **The VGG-TL2 achieves better performance than all other baseline comparisons, which provides about 71% accuracy and 0.86 AUC value**. Compared with FA and HHG methods, although the VGG16-TL1, VGG16-TL2 and VGG16-DL provide much better performances when predictions on augmented patient slides are excluded, their performances are still poorer than our proposed technique. Our proposed technique achieves around 72% accuracy and 0.89 AUC value when SVM classifiers with Gaussian kernels are used. The superior performance for the proposed technique over baseline deep learning models is mainly because our prediction models are built at patient level, which alleviates the problem of unbalanced dataset and the issue of building models with small tiles that are not guaranteed to have the same label with the WSI [30].

TABLE 3
Comparison of three-class prediction performance without augmented patient slides

Techniques	ACC (%)	AUC
FA [9]	54.95 (53.88)	0.751 (0.737)
HHG [25]	58.77 (61.28)	0.793 (0.816)
HHG+PCA	61.49 (61.35)	0.819 (0.820)
VGG16-TL1	68.27	0.815
VGG16-TL2	70.83	0.859
VGG16-DL	68.59	0.846
Proposed	72.24 (70.28)	0.898 (0.886)

3.6 Parameter tuning

There is no standard rule of thumb to determine an exact number of principal components that should be used after PCA transformation [31]. To select the best number of PCA components, we evaluate the technique with a number of empirically selected PCA components. Fig. 12(a) shows the evaluations of our method on three-class distinctions that include or exclude predictions on augmented low risk patient slides. As observed in Fig. 12(a), the performances are increasing steadily with PCA components increased from 30 to 50. Our technique achieves the highest accuracy for three-class distinctions when 50 PCA components are selected. The performances are then slightly fluctuated when the number of PCA components are increased to 70. It could be concluded from Fig. 12(a) that selecting PCA principal components within a reasonable range (e.g., from 40 to 70 in this study) can achieve good performances using the proposed technique on our evaluated dataset.

Fig. 12(b) shows the kernel scale tuning of the SVM classifier used by our technique, where kernel scale k_s is increased from 3 to 11 with a step of 2. As observed in Fig. 12(b), the SVM classifier with Polynomial kernel is relatively more robust with tested kernel scale values, where ACC values fluctuate slightly within 3%. However, the SVM classifier with Gaussian kernel is more sensitive to k_s

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2941195, IEEE/ACM Transactions on Computational Biology and Bioinformatics

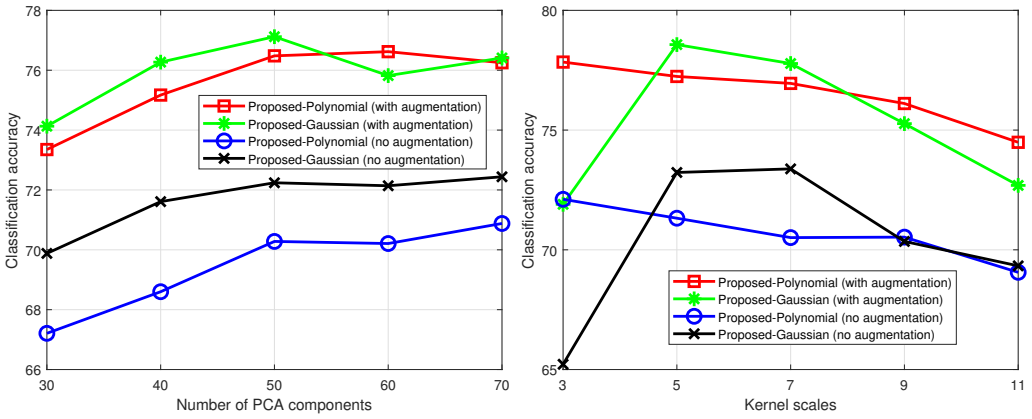

Fig. 12. Parameter tuning for PCA and SVM kernel. (a) Parameter tuning on selecting different number of PCA components. (b) Parameter tuning on using different kernel scales for SVM classifiers.

settings, which provides the lowest accuracy for different testings when k_s=3. Therefore, a suitable k_s value for the Gaussian kernel is important for achieving a good classification performance.

3.7 Evaluations with different classifiers

We additionally test several classification methods with our proposed CSLBP descriptors, which includes two SVM based binary decomposition methods and three multi-class classification methods. First, we replace the "one-against-one" multi-class SVM with "one-against-all" multi-class SVM to perform prediction. Next, in stead of using PCA for feature dimension reduction, we test to use Recursive Feature Elimination (RFE[1]) method [32], [33] with "one-against-one" multi-class SVM to perform prediction, where a subset of 50 features are selected by the RFE method during cross-valiations. We further test another direct multi-class classification method: M-SVM[2], which is an extension of the SVM formulation for binary problems [35]. Finally, we test K-Nearest-Neighbors (KNN) and bagged trees (implemented using MATLAB classification learner) with our proposed texture features. Table 4 lists classification results of both including and excluding predictions on augmented (AUG) low risk patients by using different classification methods. Note that for all SVM-based classifications, Gaussian kernels are utilized for evaluation. Since the KNN only gives hard label predictions, AUC values are not available. As observed in Table 4, compared with other texture descriptors and baseline deep learning models (see Tables 2 and 3), using different classification methods with our proposed texture features still tend to provide superior performances, which indicates the efficacy of CSLBP descriptors in encoding image textural patterns.

3.8 Computational complexity

Our proposed pipeline was evaluated on a 3.50 GHz Intel Core i7-7800 CPU with 64-GM RAM using Matlab R2018a.

1. The code is available at: https://github.com/Pegahka/
2. We used MSVMpack [34] package, which is available at: https://members.loria.fr/FLauer/files/MSVMpack/MSVMpack.html

TABLE 4
Performance of different classification methods using the CSLBP descriptors

Techniques	Including-AUG		Excluding-AUG	
	$ACC(\%)$	AUC	$ACC(\%)$	AUC
One-vs-all SVM	76.24	0.904	70.45	0.872
RFE+SVM	73.15	0.872	66.70	0.825
M-SVM	73.42	0.873	67.06	0.826
KNN	67.56	NA	58.45	NA
Bagged Trees	71.99	0.871	65.01	0.823

To train the prediction model using two thirds of patient cases, our method takes about 3 hours in total. Due to different size of pathology slides, the testing execution time for different WSI varies from each other. For example, it takes about 30 seconds in total to make predictions for the WSI with a size of 97,608×27,434×3 pixels. Specially, the image preprocessing module (see Fig. 2) takes about 4.76 seconds. The feature extraction module takes about 24.1 seconds. The image classification module with the trained classifier takes only about 1 second. In comparison, FA and HHG techniques take similar time scales as the proposed method for training and testing. For transfer learning and deep learning methods, they take much longer time for training, especially VGG16-DL which takes about 3 days for training (on a Linux server with Intel(R) Xeon(R) and GTX 1080 Ti GPU).

4 Conclusions

In this paper, we present an automatic technique for prostate cancer Gleason grading from H&E stained whole slide pathology images. The technique first divides the whole slide image into a series of image blocks for feature analysis. A set of textural features based on our proposed CSLBP descriptors are then computed from every selected image block and statistically integrated together across all image blocks to form a feature representation for a given patient slide. Finally the multi-class SVM classifier is trained on

patient level and applied to predict patient pathology slide into different risk groups corresponding to different Gleason scores. The proposed technique has been evaluated on 312 patients selected from the publicly available TCGA dataset, and experimental results show better performance over other widely-used texture descriptors and baseline deep learning models. Despite the better performance for Gleason score prediction, there exist the limitation of our proposed method, which is the use of a set of handcrafted features rather than learning and extracting features from the WSI. Therefore, as future work, we plan to extend our approach to integrate image features derived from deep learning methods with the CSLBP descriptors to assess whether integrated features from different approaches including deep learning methods could improve Gleason score prediction performance.

ACKNOWLEDGMENTS

The authors would like to thank Jean René Clemenceau, and Tyler Coy at Hwang Lab, Cleveland Clinic, for kindly going through our paper and providing valuable suggestions.

REFERENCES

[1] A. Abeshouse, J. Ahn, R. Akbani, A. Ally, S. Amin, C. D. Andry, M. Annala, A. Aprikian, J. Armenia, A. Arora *et al.*, "The molecular taxonomy of primary prostate cancer," *Cell*, vol. 163, no. 4, pp. 1011–1025, 2015.

[2] J. Gordetsky and J. Epstein, "Grading of prostatic adenocarcinoma: current state and prognostic implications," *Diagnostic pathology*, vol. 11, no. 1, p. 25, 2016.

[3] K. Nguyen, A. Sarkar, and A. K. Jain, "Prostate cancer grading: use of graph cut and spatial arrangement of nuclei," *IEEE transactions on medical imaging*, vol. 33, no. 12, pp. 2254–2270, 2014.

[4] M. K. K. Niazi, K. Yao, D. L. Zynger, S. K. Clinton, J. Chen, M. Koyutürk, T. LaFramboise, and M. Gurcan, "Visually meaningful histopathological features for automatic grading of prostate cancer," *IEEE journal of biomedical and health informatics*, vol. 21, no. 4, pp. 1027–1038, 2017.

[5] H. Källén, J. Molin, A. Heyden, C. Lundström, and K. Åström, "Towards grading gleason score using generically trained deep convolutional neural networks," in *IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2016, pp. 1163–1167.

[6] O. J. del Toro, M. Atzori, S. Otálora, M. Andersson, K. Eurén, M. Hedlund, P. Rönnquist, and H. Müller, "Convolutional neural networks for an automatic classification of prostate tissue slides with high-grade gleason score," in *SPIE Medical Imaging 2017*, vol. 10140. International Society for Optics and Photonics, 2017, p. 101400O.

[7] E. Arvaniti, K. S. Fricker, M. Moret, N. Rupp, T. Hermanns, C. Fankhauser, N. Wey, P. J. Wild, J. H. Rueschoff, and M. Claassen, "Automated gleason grading of prostate cancer tissue microarrays via deep learning," *Scientific reports*, vol. 8, 2018.

[8] D. Komura and S. Ishikawa, "Machine learning methods for histopathological image analysis," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 34–42, 2018.

[9] P.-W. Huang and C.-H. Lee, "Automatic classification for pathological prostate images based on fractal analysis," *IEEE transactions on medical imaging*, vol. 28, no. 7, pp. 1037–1050, 2009.

[10] P. Khurd, C. Bahlmann, P. Maday, A. Kamen, S. Gibbs-Strauss, E. M. Genega, and J. V. Frangioni, "Computer-aided gleason grading of prostate cancer histopathological images using texton forests," in *IEEE International Symposium on Biomedical Imaging (ISBI): From Nano to Macro*. IEEE, 2010, pp. 636–639.

[11] H. Xu, R. Berendt, N. Jha, and M. Mandal, "Automatic measurement of melanoma depth of invasion in skin histopathological images," *Micron*, vol. 97, pp. 56–67, 2017.

[12] A. Goode, B. Gilbert, J. Harkes, D. Jukic, and M. Satyanarayanan, "Openslide: A vendor-neutral software foundation for digital pathology," *Journal of pathology informatics*, vol. 4, 2013.

[13] H. Xu, C. Lu, R. Berendt, N. Jha, and M. Mandal, "Automated analysis and classification of melanocytic tumor on skin whole slide images," *Computerized Medical Imaging and Graphics*, vol. 66, pp. 124–134, 2018.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[15] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[17] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, "Median robust extended local binary pattern for texture classification," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1368–1381, 2016.

[18] L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, "Local binary features for texture classification: taxonomy and experimental study," *Pattern Recognition*, vol. 62, pp. 135–160, 2017.

[19] J. Cheng, X. Mo, X. Wang, A. Parwani, Q. Feng, and K. Huang, "Identification of topological features in renal tumor microenvironment associated with patient survival," *Bioinformatics*, vol. 1, p. 7, 2017.

[20] S. Kothari, J. H. Phan, T. H. Stokes, and M. D. Wang, "Pathology imaging informatics for quantitative analysis of whole-slide images," *Journal of the American Medical Informatics Association*, vol. 20, no. 6, pp. 1099–1108, 2013.

[21] W. L. Martinez, A. R. Martinez, and J. Solka, *Exploratory data analysis with MATLAB*. Chapman and Hall/CRC, 2017.

[22] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, NY, USA:, 2001, vol. 1, no. 10.

[23] N. Cristianini, J. Shawe-Taylor *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[24] N. Sarkar and B. Chaudhuri, "An efficient differential box-counting approach to compute fractal dimension of image," *IEEE Transactions on systems, man, and cybernetics*, vol. 24, no. 1, pp. 115–120, 1994.

[25] J. N. Kather, C.-A. Weis, F. Bianconi, S. M. Melchers, L. R. Schad, T. Gaiser, A. Marx, and F. G. Zöllner, "Multi-class texture analysis in colorectal cancer histology," *Scientific reports*, vol. 6, p. 27988, 2016.

[26] D. Fehr, H. Veeraraghavan, A. Wibmer, T. Gondo, K. Matsumoto, H. A. Vargas, E. Sala, H. Hricak, and J. O. Deasy, "Automatic classification of prostate cancer gleason scores from multiparametric magnetic resonance images," *Proceedings of the National Academy of Sciences*, vol. 112, no. 46, pp. E6265–E6273, 2015.

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

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[29] M. W. Lafarge, J. P. Pluim, K. A. Eppenhof, P. Moeskops, and M. Veta, "Domain-adversarial neural networks to address the appearance variability of histopathology images," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 83–91.

[30] 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*, 2016, pp. 2424–2433.

[31] L. Cao, K. S. Chua, W. Chong, H. Lee, and Q. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1-2, pp. 321–336, 2003.

[32] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389–422, Jan 2002.

[33] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sensors and Actuators B: Chemical*, vol. 212, 06 2015.

[34] F. Lauer and Y. Guermeur, "MSVMpack: a multi-class support vector machine package," *Journal of Machine Learning Research*, vol. 12, pp. 2269–2272, 2011.
[35] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," Technical Report CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science, Tech. Rep., 1998.

Hongming Xu is currently a postdoc fellow working at Cleveland Clinic, Cleveland, USA. He received his PhD degree from Department of Electrical and Computer Engineering at University of Alberta, Edmonton, Canada, in 2017. He received his B.Sc. and M.Sc. degrees from Information Engineering College at Northwest A&F University, Shaanxi, China, in 2009 and 2012. His research interest spans over machine learning, deep learning, computer vision and medical image analysis. His current projects focus on exploring state-of-the-art AI techniques for pathology imaging to predict cancer patient clinical outcomes. He has authored and published over 15 referred journal and conference papers.

Sunho Park received the bachelor's degree in Electrical, Electronics and Radiowave engineering from Korea University, Seoul, Korea in 2004, and the master and Ph.D. degrees in Computer Science from Pohang University of Science and Technology, Korea in 2013. He was a postdoctoral researcher in the department of Clinical Sciences at University of Texas Southwestern Medical Center from 2013 to 2017, and have been a postdoctoral researcher in the department of Quantitative Health Sciences at Cleveland clinic. His main areas of research interests are Bayesian matrix factorization, Gaussian process and its variants, including deep Gaussian process, and machine learning applications in biomedical areas.

Tae Hyun Hwang is an assistant professor of Molecular Medicine at Cleveland Clinic Lerner College of Medicine. He received his PhD in Computer Science at the University of Minnesota Twin-Cites at 2011. He and his research group lead machine learning and AI research at Cleveland Clinic. Prior his appointment at the Cleveland Clinic, he was a tenure-track faculty at the University of Texas Southwestern Medical Center where he led a team of computational scientists for cancer research. He currently serves as a bioinformatics core director for NASA Specialized Centers of Research (NSCOR) as well as committees of various Machine Learning, Data Mining, and AI conferences.