

Automated proliferation index calculation for skin melanoma biopsy images using machine learning

Salah Alheejawi^a, Richard Berendt^b, Naresh Jha^b, Santi P. Maity^c, Mrinal Mandal^{a,*}

^a Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, T6G 1H9, Canada

^b Department of Medicine, University of Alberta, Edmonton, Alberta, T6G 2B7, Canada

^c Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, 711103, India

ARTICLE INFO

Keywords:

Histopathological image analysis
Nuclei segmentation
Proliferation index calculation
Melanoma
Machine learning

ABSTRACT

The Proliferation Index (PI) is an important diagnostic, predictive and prognostic parameter used for evaluating different types of cancer. This paper presents an automated technique to measure the PI values for skin melanoma images using machine learning algorithms. The proposed technique first analyzes a Mart-1 stained histology image and generates a region of interest (ROI) mask for the tumor. The ROI mask is then used to locate the tumor regions in the corresponding Ki-67 stained image. The nuclei in the Ki-67 ROI are then segmented and classified using a Convolutional Neural Network (CNN), and the PI value is calculated based on the number of the active and the passive nuclei. Experimental results show that the proposed technique can robustly segment (with 94 % accuracy) and classify the nuclei with a low computational complexity and the calculated PI values have less than 4 % average error.

1. Introduction

Skin cancer is the most common cancer type in North America, and it can be of different types such as Merkel cell carcinoma, sebaceous carcinoma, basal cell carcinoma and melanoma. Among these, malignant melanoma is the deadliest type with high mortality as it can spread to any part of the body. As per recent statistics, it is estimated that about 100,350 new cases of cutaneous malignant melanoma (CMM) cancer would be diagnosed resulting in about 6,850 deaths in 2020 in the United States alone (Siegel et al., 2020). The CMM occurs when the melanocytes, located around the junction of the epidermis and the dermis layers in the skin, start growing aggressively to form metastatic melanoma. The aggressive growth of melanocytes is mainly caused by the DNA replication due to the excessive exposure of the ultraviolet radiation (emitted from the sun). Once it is established that melanoma has invaded the epidermis, dermis and subcutaneous layer (in skin), doctors examine the lymph nodes closest to the tumor, known as the Sentinel Lymph Nodes (SLNs) to determine the spread. Through lymph vessels, the metastatic melanoma can be drained from the skin to the SLNs and is passed on to other organs. The early detection, diagnosis, and staging of melanoma can increase the survival rate up to 92 % after five years when the cancer is found (Hanahan and Weinberg, 2011).

The proliferative activity of CMM is indicative of neoplastic progression and often the clinical growth rate of the neoplasm. High risk CMM is commonly associated with a higher growth rate, which is reflected in a higher Proliferation Index (PI) of the tumor regions. The PI as measured by Ki67 expression, reflects the degree of active cell proliferation (Bánkfalvi, 2000), and may be useful pathological predictor of survival of melanoma patients without distant metastases. PI as a predictor of tumor progression can help physicians to determine the effective treatment (Nielsen et al., 2013). The PI is determined by estimating the ratio of the number of active cell nuclei and the total number of cell nuclei. During cell proliferation, the cell nuclei become active and start replicating the nuclei DNA and cytoplasm. A stained skin tissue slide is digitized and diagnosed by a pathologist. In Ki-67 stained image, the positive response of the actively proliferative cells to the Ki-67 marker appears as dark brown color, whereas other cells appear as blue color (see Fig. 1). A pathologist usually chooses the high proliferative areas in the tumor regions and manually calculates the PI values by counting the proportion of the active cells. The manual detection of the tumor regions, finding the high proliferative areas and calculating the PI values in a Whole Slide Image (WSI) can be labor and time intensive. In addition to the large size of WSIs, the manual analysis is also subjected to intra- and inter-observer variability (Reid et al., 2015). Therefore,

* Corresponding author.

E-mail addresses: alheejaw@ualberta.ca (S. Alheejawi), richard.berendt@albertahealthservices.ca (R. Berendt), naresh.jha@albertahealthservices.ca (N. Jha), santipmaity@it.iests.ac.in (S.P. Maity), mmandal@ualberta.ca (M. Mandal).

<https://doi.org/10.1016/j.compmedig.2021.101893>

Received 14 September 2020; Received in revised form 5 January 2021; Accepted 25 February 2021

Available online 5 March 2021

0895-6111/© 2021 Elsevier Ltd. All rights reserved.

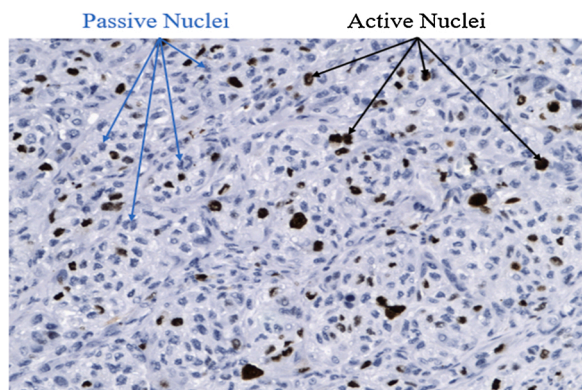


Fig. 1. Histopathological image stained with Ki-67 stain.

automated Computer-Aided Diagnosis (CAD) algorithms can provide a fast and reliable diagnosis (Wang et al., 2009).

A few techniques have been proposed in the literature to calculate the PI values in Ki-67 stained images for different types of cancers. The PI calculation is generally done by segmenting and/or detecting the nuclei (both active and passive nuclei). The nuclei segmentation and detection are difficult in Ki-67 images, as most cells lose the blob shape and have heterogeneity in color representation. To calculate PI values, several researchers used thresholding algorithms for cells segmentation followed by a classifier. Grala et al. (2009) proposed sequential thresholding to segment the cells and Support Vector Machine (SVM) to classify them into active and passive cells. Akakin et al. (2012) used watershed algorithm and Laplacian-of-Gaussian filters to enhance the cell segmentation. Al-Lahham et al. (2012) proposed a technique to segment and classify the cells into active/passive nuclei by applying K-means algorithm on the image pixels in $L^*a^*b^*$ color-space. Mungle and Tewary (2017) proposed an algorithm to segment the cells using a Fuzzy C-means (FCM) technique and then apply the K-means (KM) algorithm for classification (henceforth referred to as FCM + KM technique). Alheejawi et al. (2019a) proposed an automated algorithm to calculate the PI values on the tumor regions in Ki-67 images. The algorithm determined the tumor regions by mapping the segmented melanoma areas from Mart-1 onto Ki-67 image. The PI values in this technique are calculated by segmenting the cells using the Otsu threshold and then classifying the active/passive nuclei using a SVM (henceforth referred to as the Otsu + SVM technique).

With the recent advances in Machine Learning (ML) algorithms, the convolutional neural networks (CNN) have become popular for medical image segmentation and classification. Long et al. (2015) proposed a CNN architecture for the semantic segmentation with several convolutional layers and a few max pooling layers followed by one upsampling layer. This architecture shows the segmented objects with coarse boundaries/edges due to the information lost in the pooling layers. Badrinarayanan et al. (2017) proposed a CNN architecture (popularly known as SegNet) by adding several upsampling layers and enhanced them by transferring the maximum value indices of the corresponding pooling layers. Ronneberger et al. (2015) proposed a CNN architecture, popularly known as UNet, to improve medical image segmentation by concatenating the upsampling layers (at the decoder side) with the corresponding features maps from the pooling layers (at the encoder side). Chen et al. (2018) proposed a CNN architecture (known as DeepLabV3) for object segmentation, which contains several dilated separable convolutional layers, spatial pyramid pooling, and a few residual connections (Kaiming et al., 2016). Although the DeepLab architecture has been shown to provide a good segmentation performance, it has high computational complexity due to the large number of layers. In addition, pooling layers in the DeepLab may degrade the segmented object boundaries especially in medical images.

Saha et al. (2017) proposed a technique (henceforth referred to as

the GMM + CNN technique) to calculate PI values in breast cancer biopsy. Here, a few Ki-67 image windows with dense active nuclei are manually selected for the PI calculation. This technique uses a Gamma Mixture Model (GMM) with expectation-maximization algorithm to detect the nuclei seeds. Image patches around the isolated nuclei seeds are then classified into passive/active nuclei using a CNN classifier. The CNN uses 5 convolutional layers followed by two fully connected layers. In addition to the high computational complexity of the nuclei detection, the technique can calculate the PI values only for the isolated cell nuclei, which typically increases the error rate. Alheejawi et al. (2019b) proposed an automated technique (henceforth referred to as the PI-SegNet technique) to calculate the PI values for the melanoma regions using SegNet architecture, which contains 8 convolutional, 2 pooling, and 2 un-pooling layers followed by a softmax layer. The pooling layers can degrade the segmentation performance.

In this paper, a fully automated CNN-based technique is proposed to calculate the PI value in a WSI. The proposed technique first generates a region of interest (ROI) mask in a Mart-1 image based on color histogram analysis, and the mask is mapped onto the corresponding Ki-67 image to identify the melanoma region. The nuclei in the Ki-67 image are then segmented and classified using a CNN architecture, and PI values are calculated based on the proportion of active and passive nuclei areas. Experimental results demonstrate high accuracy and low error rate of the calculated PI values compared to the state-of-the-art techniques.

The organization of the paper is as follows. Section 2 describes the dataset used to train and evaluate the proposed algorithm; Section 3 describes the proposed technique in detail. The performance evaluation of the proposed technique is presented in Section 4 followed by the conclusion in Section 5.

2. Data description

In this study, 10 pairs of histopathological images (from 10 different patients), obtained using Mart-1 and Ki-67 stains, were collected from the Cross-Cancer Institute, University of Alberta, Canada in accordance with the protocol for the examination of patients with melanoma cancer. The histopathological slides were digitized under 40X magnification using aperio scanner. The obtained WSIs are with size around $60,000 \times 31,000$ pixels (15 mm x 7.75 mm area with $0.25 \mu\text{m}/\text{pixel}$ resolution). Each WSI contains several lymph nodes, and some of these lymph nodes contain 1-4 melanoma regions.

Seven non overlapping windows (i.e., ROIs) with size 704×1280 color pixels are obtained from the lymph node regions of each Ki-67 stained WSI. In other words, a total of 70 windows are obtained from 10 WSIs, which form the dataset for performance evaluation.

As the dataset is small, five-fold cross validation is used for performance evaluation in this paper. For each fold, two WSIs are arbitrarily chosen as the testing WSIs, and 14 ROI windows from these two WSIs are used as the testing dataset. The remaining 56 ROI windows (from the other 8 WSIs) are used as the training dataset. This process is repeated five times for the five-fold cross validation, and five sets of performance measures are obtained. The average performance is reported in Section 4

3. Proposed technique

The schematic of the proposed technique is shown in Fig. 2. The proposed technique consists of three modules: Melanoma Region Identification, CNN-based segmentation and PI calculation. The details of each module are presented in the following.

3.1. Tumor region identification

The PI values are calculated only within the melanoma regions in the Ki-67 images. The objective of this module is to identify the melanoma regions in Ki-67 stained lymph node image with the help of Mart-1

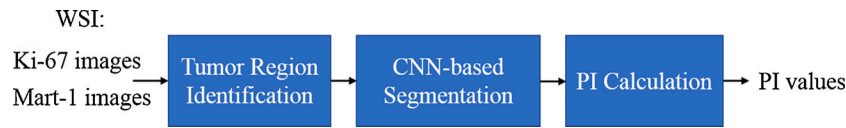


Fig. 2. Schematic of the proposed segmentation technique.

stained image of a consecutive section (Alheejawi et al., 2019a). The melanoma regions appear in brown color in Mart-1 stain. These regions are segmented and superimposed on Ki-67 images to identify the tumor regions in Ki-67 images. Fig. 3(a) shows two lymph nodes in Mart-1 and the melanoma regions appear in brown color. Fig. 3(b) shows the mapped melanoma regions in green contour on Ki-67 image. Let the melanoma regions on Ki-67 stained image be denoted by R_i ($i = 1, 2, \dots, N_{DR}$) where N_{DR} is the total number of disconnected melanoma regions in the lymph nodes.

3.2. CNN-based segmentation

The objective of this module is to segment an input image window (from the detected melanoma region) into active nuclei, passive nuclei and background. In this paper, a new CNN architecture, as shown in Fig. 4, is used to segment the Ki-67 images into three classes (background, passive and active nuclei). The CNN architecture consists of three paths: path A, path B and path C. The details of these three paths are listed in Table 1. The paths A and B have 8 and 9 convolutional layers, respectively (see Fig. 4). Path C provides the residual connection (also known as skip connection) from the input layer to the Concatenate layer (denoted as Layer-CON in Table 1). The three paths start after the first convolution layer, become parallel, and meet again at the concatenate layer. This is followed by a convolutional layer, softmax and pixel classification layer. The details of these layers are explained in the following.

1) *Convolutional layers*: The convolutional layer is similar to the neurons in a traditional neural network except that each neuron will be connected to a small neighborhood of neurons in the previous layer.

Each neuron in the convolutional layer will result in a scalar value. In this paper, the convolutional layer performs three operations: convolution, batch normalization, and ReLU activation.

(a) Let f_{l-1} denote the (3D) feature map generated in the convolutional layer $l-1$. In the convolution layer l , the feature map f_{l-1} is convolved with a (3D) filter F_j :

$$R_{l,j} = f_{l-1} * F_j, \quad j = 1, 2, \dots, N \quad (1)$$

where N is the number of filters in layer l (also known as the depth of the layer l), $R_{l,j}$ is the (2D) output corresponding to the j th convolution filter. Note that for the first convolution layer (i.e., $l = 1$), the input image is considered as f_0 . In the first convolutional layer (Layer 1) of the network in Fig. 4, f_0 , F_j , and $R_{1,j}$ have dimensions of $H \times W \times 3$, $3 \times 3 \times 3$, and $H \times W \times 64$, respectively (with the input image having $H \times W$ pixels). In the proposed CNN, the number of filters is 64/layer (in both paths A and B) with size of $S \times S$ (see Table 1). Note that S increases from 5 to 21 in subsequent layers in path B. However, it has a fixed value of 3 in all layers of path A. Note that the paths A and B provide complementary features: the path A focuses more on the coarse features whereas the path B focuses more on the fine features.

(b) *Batch Normalization*: During the CNN training, the convolution outputs $R_{l,j}$ corresponding to all images in a mini-batch (of B images) are considered. In this work, we have used $B = 8$ (Ioffe and Szegedy, 2015).

The $R_{l,j}$ is made zero mean with unit variance as follows:

$$\hat{R}_{l,j} = \frac{R_{l,j} - \mu_j}{\sigma_j} \quad (2)$$

where μ_j and σ_j are the mean and the standard deviation corresponding

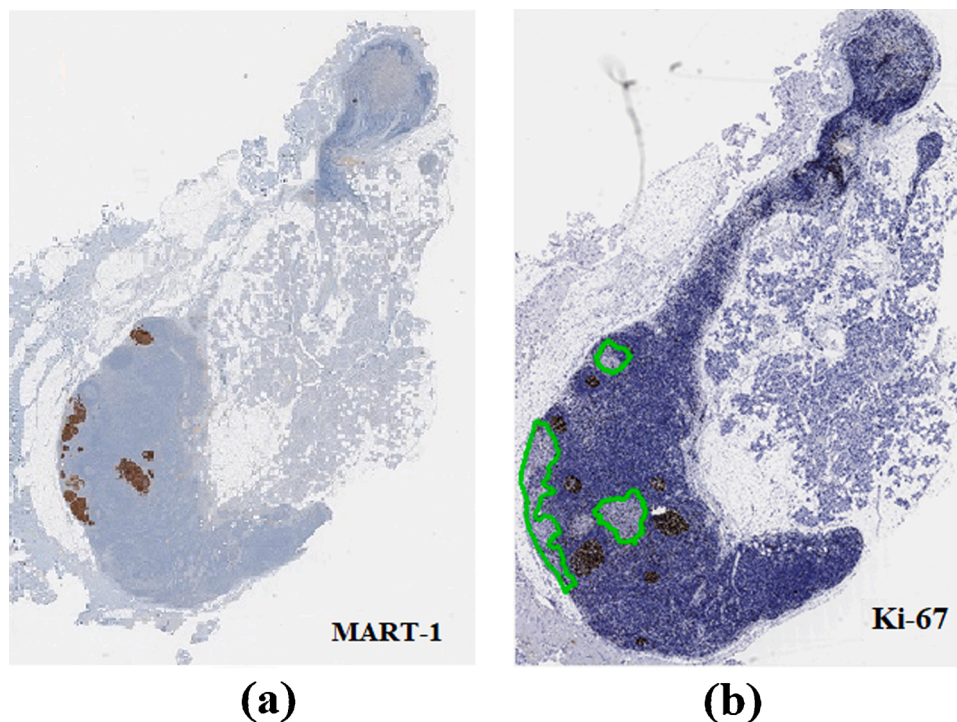


Fig. 3. Histopathological image of two lymph nodes (a) Mart-1 stained image, where melanoma appears in brown color and (b) Ki-67 stained image, where the green contour is the melanoma mask. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

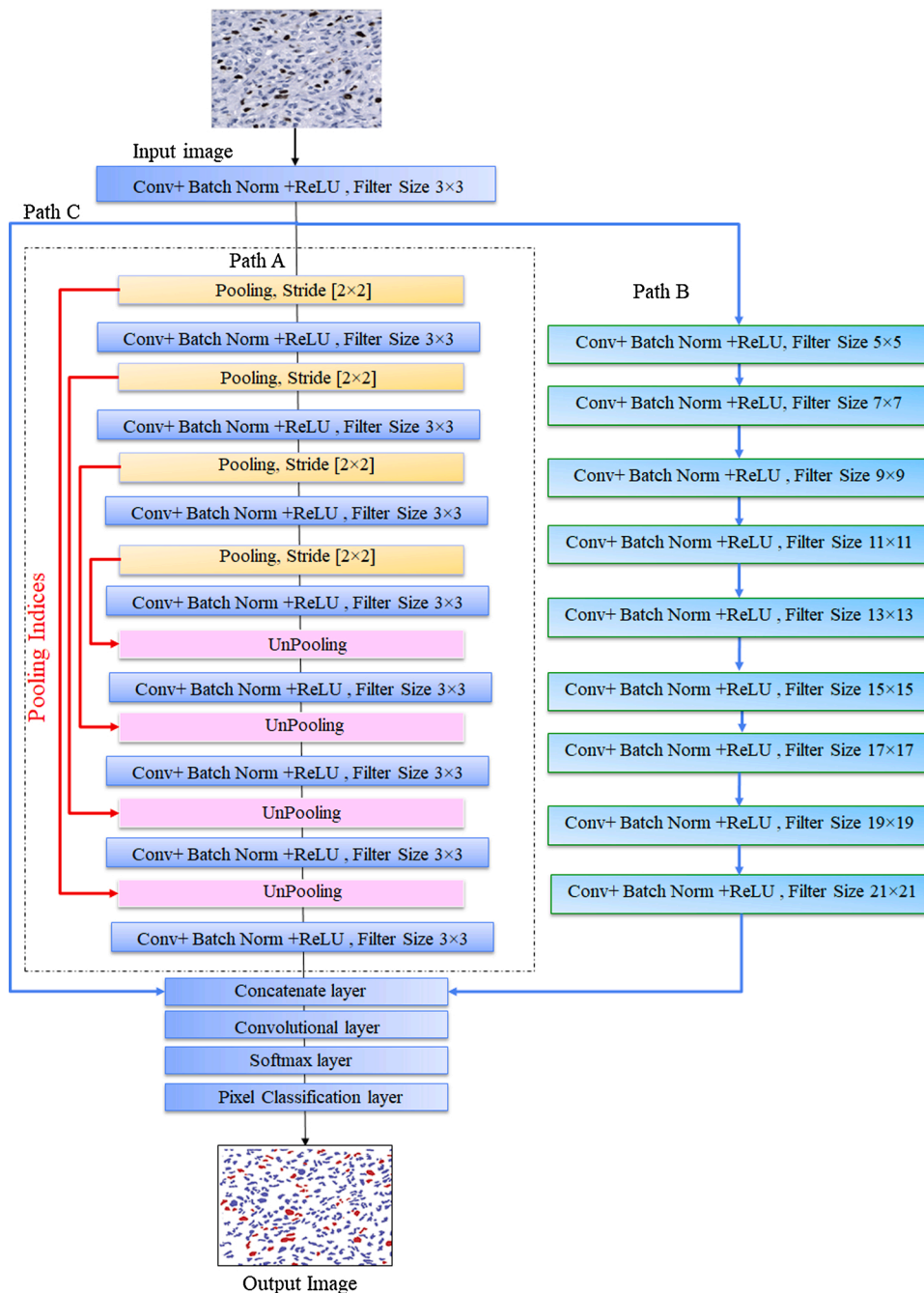


Fig. 4. Proposed CNN architecture for nuclei segmentation (with 19 convolutional layers). Note that there are three paths after Layer I (first convolutional layer). Path A has 8 convolutional (shown in blue), 4 pooling (yellow) and 4 un-pooling (pink) layers. Path B has 9 convolutional layers. Path C is the residual connection from the output of layer I to the concatenate layer. Paths A, B and C meet at the Concatenate Layer. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

to the j th feature map for the layer l in a mini-batch. The normalized $\hat{R}_{l,j}$ is scaled with γ , and a bias β is added as follows:

$$y_{l,j} = \gamma \hat{R}_{l,j} + \beta_{l,j} \quad (3)$$

Note that Eqs. 2 and 3 are applied during both the training (with $B = 8$) and the testing (with $B = 1$). The γ and β are trainable parameters and are updated iteratively during the backpropagation.

(c) Activation: In this step, an activation function is applied on the batch normalized output $y_{l,j}$. In this work, the Rectified Linear Unit (ReLU) activation (Nair and Hinton, 2010) is used as it has the advantages of speeding up the training (Krizhevsky et al., 2012) and inducing the sparsity in the hidden units (Glorot et al., 2011). In ReLU, all the negative pixel values are replaced by zeros, and the output is expressed as follows.

$$f_{l,j} = \max(0, y_{l,j}) \quad (4)$$

The overall output of convolutional layer l , as given below, will be passed on to the next layer.

$$f_l = \{f_{l,j}, j = 0, 1, \dots, N\}$$

Step (a)-(c) are repeated for each subsequent convolutional layer. After the ReLU module, the feature map f_l is used as the input of the next, i.e., the $(l+1)$ th, convolutional layer.

2) *Pooling/ UnPooling Layers*: The pooling layers are used to extract the features in different image resolution, whereas the unpooling layers are used to return the extracted features with the same size as the original image ($H \times W$) using bilinear interpolation. In the pooling layers, the max-pooling operations for $[2 \times 2]$ pixels with stride of 2x2 are used. The indices of the maximum values in pooling layers are

Table 1

Details of the proposed architecture with 19 convolutional layers. Input image Size: $H \times W$ pixels (color). C: Number of classes, NA: Not Applicable. Note that layers A1,A3,A5,A7 are pooling layers, and layers-A9,A11,A13,A15 are UnPooling layers. For these layers, the max-pooling and a stride of 2x2 are used.

	Number of Channels (N)	Number of Filters	Output Image Size	Filter Size (S)
Layer-I	3	64	$H \times W \times 64$	3×3
Layer-A1	64	NA (Pooling)	$(H/2) \times (W/2) \times 64$	-
Layer-A2	64	64	$(H/2) \times (W/2) \times 64$	3×3
Layer-A3	64	NA (Pooling)	$(H/4) \times (W/4) \times 64$	-
Layer-A4	64	64	$(H/4) \times (W/4) \times 64$	3×3
Layer-A5	64	NA (Pooling)	$(H/8) \times (W/8) \times 64$	-
Layer-A6	64	64	$(H/8) \times (W/8) \times 64$	3×3
Layer-A7	64	NA (Pooling)	$(H/16) \times (W/16) \times 64$	-
Layer-A8	64	64	$(H/16) \times (W/16) \times 64$	-
Layer-A9	64	NA (UnPooling)	$(H/8) \times (W/8) \times 64$	-
Layer-A10	64	64	$(H/8) \times (W/8) \times 64$	3×3
Layer-A11	64	NA (UnPooling)	$(H/4) \times (W/4) \times 64$	-
Layer-A12	64	64	$(H/4) \times (W/4) \times 64$	3×3
Layer-A13	64	NA (UnPooling)	$(H/2) \times (W/2) \times 64$	-
Layer-A14	64	64	$(H/2) \times (W/2) \times 64$	3×3
Layer-A15	64	NA (UnPooling)	$H \times W \times 64$	-
Layer-A16	64	64	$H \times W \times 64$	3×3
Layer-B1	64	64	$H \times W \times 64$	5×5
Layer-B2	64	64	$H \times W \times 64$	7×7
Layer-B3	64	64	$H \times W \times 64$	9×9
Layer-B4	64	64	$H \times W \times 64$	11×11
Layer-B5	64	64	$H \times W \times 64$	13×13
Layer-B6	64	64	$H \times W \times 64$	15×15
Layer-B7	64	64	$H \times W \times 64$	17×17
Layer-B8	64	64	$H \times W \times 64$	19×19
Layer-B9	64	64	$H \times W \times 64$	21×21
Layer-CON	192	NA (Concatenate)	$H \times W \times 192$	-
Layer-O	64	C	$H \times W \times C$	3×3
Softmax	C	NA	$H \times W \times C$	-
Pix-Class	1	NA	$H \times W$	-

transferred to the corresponding unpooling layers.

3) Concatenate Layer: In this layer, the outputs of paths A, B and C, each with size $H \times W \times 64$, are concatenated to obtain an output feature map of size $H \times W \times 192$.

4) *Softmax*: The softmax layer is used after the last convolutional layer to generate the probability distribution of the classification results for each pixel. The output of the last convolutional layer consists of C feature maps (each of size $H \times W$), where C is the number of the output classes. Let the feature map corresponding to the i th class ($i = 1, 2, \dots, C$) be denoted by ϕ_i . The exponential summation of the feature maps, Φ , is calculated as follows:

$$\Phi(h, w) = \sum_{i=1}^C \exp(\phi_i(h, w)) \quad 1 \leq h \leq H, 1 \leq w \leq W \quad (5)$$

The softmax layer calculates the class probability P_i for the pixel at location (h, w) using the following equation.

$$P_i(h, w) = \frac{\exp(\phi_i(h, w))}{\Phi(h, w)} \quad (6)$$

Note that the input pixels are classified into three classes (i.e., $C = 3$): background, active nuclei and passive nuclei.

5) *Pixel classification*: This layer performs the classification of the image pixels using the corresponding class probability. The class with the highest probability is chosen as the pixel output class.

3.3. CNN training

There are about 7,513,247 parameters in the proposed CNN, which need to be trained. In the training phase of the CNN, 56 ROI windows (as described in section 2) are used. Note that each window contains thousands of nuclei (see Fig. 5). These 56 windows are divided into training (49 windows) and validation (7 windows) datasets. As it is computationally expensive to use the whole window as input to the CNN, each window is further divided into non-overlapping blocks of 64×64 color pixels to obtain 220 block-images (an example of a block-image is shown in Fig. 5). The total number of the obtained block-images is 12,320 ($= 56 \times 220$).

The 12,320 images are also augmented by horizontal and vertical flipping to obtain 36,960 images (32,340 training and 4,620 for validation). The image augmentation is used to generalize the parameters of the CNN model and have better performance (Shijie et al., 2017).

For CNN training, a loss function and an optimization algorithm need to be defined. In this paper, we use the cross-entropy loss function for multi-class classification. The loss function X for a pixel at (m, n) location is calculated as follows:

$$X(m, n) = \sum_{i=1}^C T_i(m, n) \ln P_i(m, n) \quad (7)$$

where $T_i(m, n)$ is the i th-ground truth class probability and $P_i(m, n)$ is the i th-class probability (predicted by the Softmax layer). The total loss for a mini-batch of B block-images (each with size $H \times W$) is then calculated

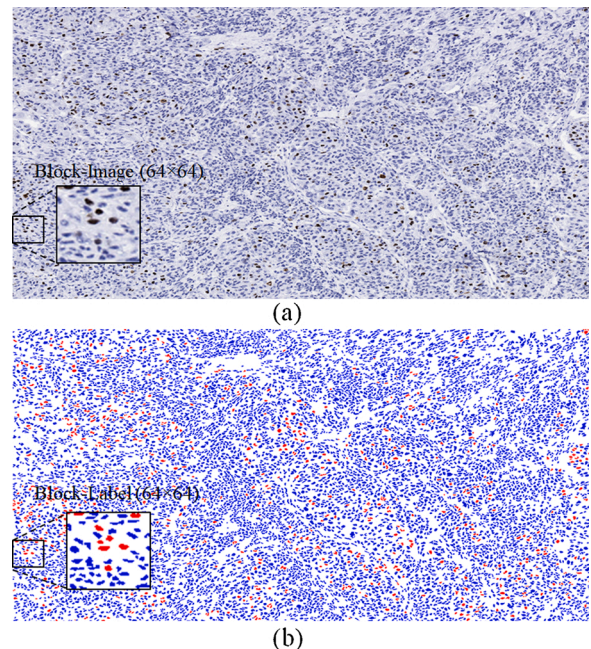


Fig. 5. Example of (a) an original RGB Image (704×1280) and (b) the corresponding classified ground truth image, where the active and passive nuclei appear in red and blue color, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

as follows:

$$E = \sum_{b=1}^B \sum_{m=1}^H \sum_{n=1}^W X_b(m, n) \quad (8)$$

where X_b is the loss function for the b th-block image. As the three classes of pixels have different distributions for each class, the loss function is weighted for each class differently. The weights φ_i for different classes are calculated as follows:

$$\varphi_i = \frac{\text{median freq}}{\text{freq}(i)} \quad 1 \leq i \leq C \quad (9)$$

where the class frequency $\text{freq}(i)$ is a ratio of the number of the pixels that represent the i -class to the total number of pixels in the training dataset, and median freq is the median of the class frequencies $\{\text{freq}(i)\}$. To minimize the loss function E , the stochastic gradient descent with momentum (SGDM) optimizer (Murphy, 2012) is used in the proposed technique. The SGDM accelerates stochastic gradient descent (SGD) and reduces the oscillation problem of SGD by adding the contribution from the previous iteration to the current iteration. The network parameters at the end of the n th iteration are updated as follows:

$$\theta_{n+1} = \theta_n - \alpha \nabla E(\theta_n) + \mu(\theta_n - \theta_{n-1}) \quad (10)$$

where θ is the parameter vector, which includes parameters such as normalization weights, filters and biases, $\nabla E(\theta)$ is the gradient of the loss function, α is the learning rate, and μ is the momentum. The learning rate α is initialized to 0.01 and is reduced by a factor of 0.2 every 10 epochs. In this work, we have used $\mu = 0.9$, and the maximum number of epochs is set to 100. Note that the training accuracy and the loss are calculated for each individual mini-batch.

Fig. 6 shows an example of the evolution of the training and the validation accuracy over different iterations. The validation accuracy and the loss are calculated on the validation dataset at the end of every 250 iterations. The network training is stopped when the validation loss has not decreased in the previous 5 validations. Note that each epoch performs 4,042 (= 32,340/8) iterations and the training of the CNN stopped at iteration 28,644 (at around 7th epoch) in this example.

The CNN model is trained using 56 Ki-67 ROI windows (49 for training and 7 for validation) as described in Section 2. As five-fold cross validation is used in this paper, five CNN models are trained, one for each fold.

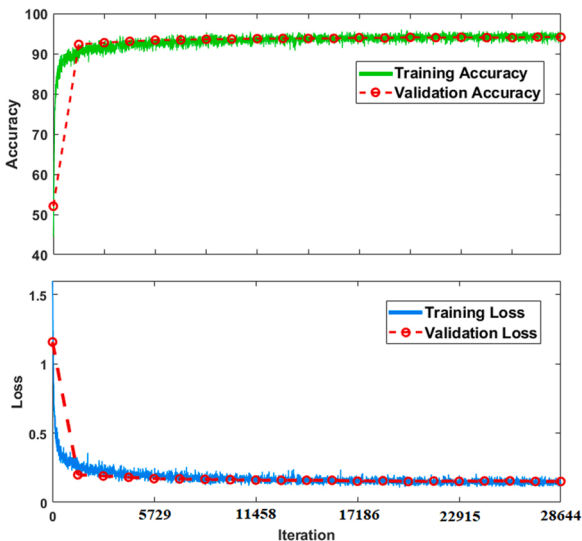


Fig. 6. Loss and accuracy plots during the network training process.

3.4. CNN evaluation

The CNN architecture proposed for the nuclei segmentation was shown in Fig. 4. In this section, we evaluate the performance of the different paths and finetune the hyper parameters to obtain the best performance. The segmentation performance is measured using: Accuracy, Intersection over Union (IoU), and *BF* score, which are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$

$$\text{BF Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP , TN , FN and FP denote the number of true positives, true negatives, false negatives and false positives, respectively. Note that the *BF* Score is defined as the harmonic mean (F1-measure) of the Precision and Recall values (calculated using the following equations) with a distance error tolerance to decide whether a point on the predicted boundary has a match on the ground truth boundary or not (Csurka et al., 2013).

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

The performance measures are calculated for each of the three pixel-classes separately. For example, when the IoU for active nuclei class is calculated, the active nuclei pixels are considered positive and both passive and background pixels are considered as negative. The performance measures for three classes are then averaged to obtain the mean accuracy, mean IoU and mean *BF* score, which are used for the overall performance comparison.

To analyze the performance of each path, an ablation study is performed, and the performance of the different combinations of paths is shown in Table 2. It is observed that the proposed architecture, with all three paths A, B and C, provides performance superior to the other combinations: A, A + B or A + C. The Path A can efficiently detect the nuclei in different scales by applying several pooling and un-pooling layers. The Path A is therefore helpful to detect the coarse features (due to pooling) of the objects. Path B has been used with different sizes of filters and without using any pooling layers to obtain fine segmentation of the nuclei. The residual connection in path C reduces the impact of vanishing gradients problem caused by the large number of layers in paths A and B (Chen et al., 2018; Zeiler and Fergus, 2014). By combining all three paths, there will be minimal information loss and better feature extraction to segment the three classes of pixels.

Fig. 7 shows visual examples of nuclei segmentation results, with and without Path B. It is observed that the CNN with path A + C provides the coarse segmentation boundaries. When the Path B is included in the CNN, more accurate segmentation results are obtained for both Passive (contoured in cyan) and Active (contoured in red) nuclei.

Fig. 8 shows the nuclei segmentation performance with different numbers of convolutional layers in path B (paths A and C are considered

Table 2
Segmentation performance of the proposed technique.

Technique:	Mean Accuracy	Mean IoU	Mean BF Score
Path A	0.86	0.80	0.70
Path A + C	0.90	0.83	0.71
Path A + B	0.91	0.85	0.74
Path A + B+C	0.94	0.89	0.82

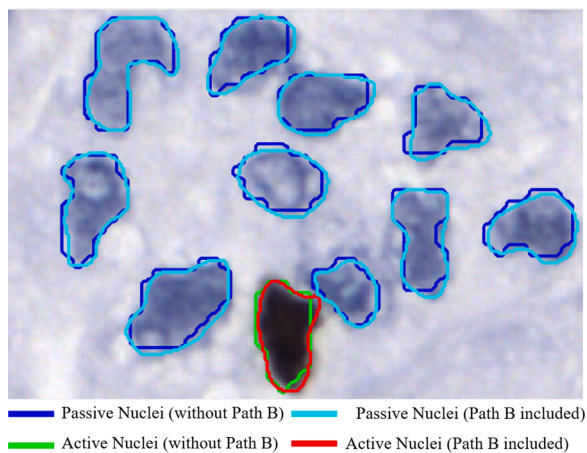


Fig. 7. Comparison of the nuclei segmentation performance using paths A + C (i.e., without path B) and paths A + B + C.

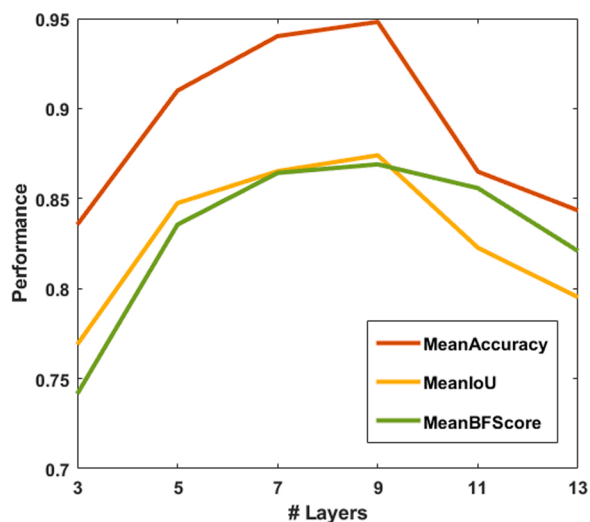


Fig. 8. Segmentation performance evaluation with different number of layers in path B.

fixed during the performance evaluation). It is observed that nine layers in path B provides the best performance. In a similar way, Path A was also evaluated (results not shown in the paper) to determine the optimal number of layers (keeping paths B and C fixed). Eight convolutional layers and four pooling and four un-pooling layers in Path A were found to provide the best nuclei segmentation performance.

Fig. 9 shows the performance of the proposed CNN with different size of filters in path B. In this evaluation, eight CNN architectures are trained with 9 layers. In the first seven architectures, the filter size for all layers is kept fixed, i.e., $S \times S$, where $S = 3, 5, 7, 9, 11, 13$ and 15 for the seven architectures. For the eighth architecture, the filter size for each layer increases from 5 to 21 in steps of 2. As observed in Fig. 9, the eighth architecture (which includes variable filter size) provides the best segmentation performance.

Fig. 10 shows an example of the obtained features maps on Path B of the proposed CNN. Fig. 10 (a) shows an input image and Fig. 10(b) shows the corresponding segmented ground truth image where the active, passive, and background pixels are shown in red, blue and white. As shown in Table 1, there are 64 feature maps at the output of convolution Layer-B3. Twelve of the 64 feature maps are shown in Fig. 10(c)–(f). Fig. 10(g) shows the three ($C = 3$) channel output of Layer-O, which primarily represents the active, passive, and background feature maps. Fig. 10(h) shows the output of the softmax layer: the three

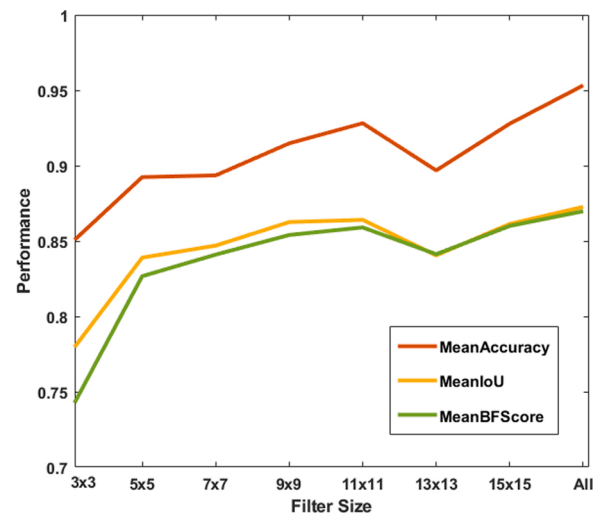


Fig. 9. Performance evaluation with different size of filters in Path B.

images showing the class probabilities (in the range $[0,1]$) of each pixel. Note that for a pixel, the sum of all three probabilities (from three images) is one. As observed in Fig. 10, there are feature maps that can represent one class as shown in (c), (d), and (e), or two as shown in (d). Note that the feature maps in (c), (d), (e), and (f) primarily detect the background, passive nuclei, active nuclei, and passive + active nuclei pixels, respectively.

3.5. PI calculation

After the melanoma tumor regions in a Ki-67 image is segmented into three classes, the next task is to determine the melanoma subregions (ROI) with the highest PI values. The PI value of an ROI is typically calculated using the following equation:

$$PI = \frac{NA}{NA + NP} \times 100 \quad (11)$$

where NA and NP are the number of the active and the passive nuclei in the ROI. In this paper, before PI values are calculated, the active and the passive nuclei are detected using the following steps.

- (i) The active and the passive nuclei pixels are determined using the CNN.
- (ii) Connected component analysis is done to form active and passive nuclei from the classified pixels. Most of the nuclei will be isolated nuclei, but some will be overlapped nuclei, also known as nuclei clump.
- (iii) Ellipse descriptor analysis is applied to separate the isolated nuclei from the nuclei clumps in the ROI (based on ellipticity parameter and size of the nuclei) (Xu et al., 2014).
- (iv) The nuclei clumps are segmented into multiple overlapped active or passive nuclei using generalized Laplacian of Gaussian (gLoG) seed detection algorithm (Xu et al., 2017).

In this paper, a few ROI windows ($K \times L$ pixels) in melanoma regions with highest PI values are determined using the following steps.

- (i) In the tumor region, a moving window ($K \times L$ pixels) is used.
- (ii) For each window, the numbers of active (NA) and passive (NP) nuclei inside the window are calculated.
- (iii) Based on the NA and NP, the PI value of a window is calculated using Eq. (11).
- (iv) A few windows with highest PI values are chosen for grading the tumor and further analysis.

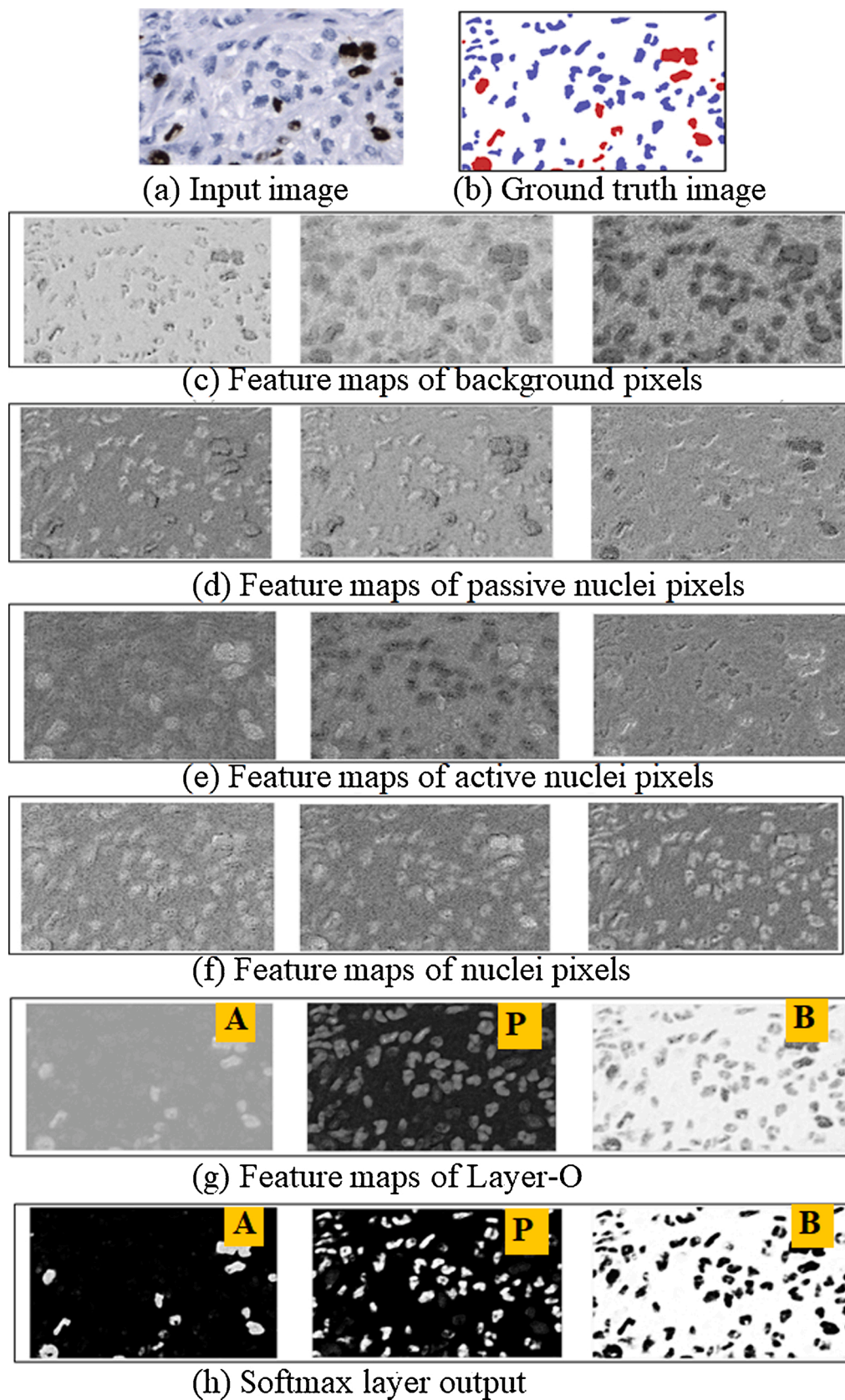


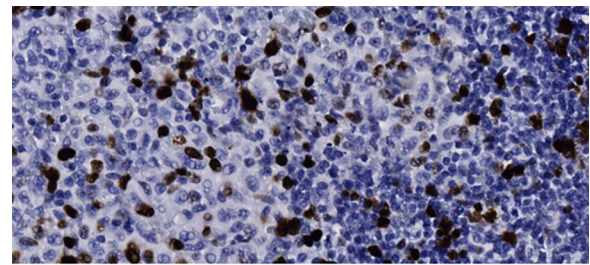
Fig. 10. Example of the feature maps obtained from path B in the proposed CNN. (a) Ki-67 image. (b) ground truth segmented image, (c)–(f) 12 of the 64 feature maps obtained after Layer-B3. (g) Output of Layer-O. (h) Softmax layer output. A: Active nuclei, P: Passive nuclei, and B: Background feature map.

Fig. 11 shows the calculation of few PI values in three melanoma regions (MR_1 , MR_2 , MR_3) in the WSI. The window size used is 250×500 ($K \times L$) pixels. Two or three windows with the highest PI values are shown in each of the three melanoma regions. Finally, the three windows with overall highest PI values are shown with red contour. A magnified version of the window with the highest PI values ($PI = 24.14$), located in MR_3 , is shown in Fig. 12. The active and the passive nuclei are shown as red and green + symbols, respectively. The overall nuclei (active and passive) classification accuracy is about 94 %. For the window shown in Fig. 12, the ground truth PI value is 25.15 while the obtained PI value is 24.14.

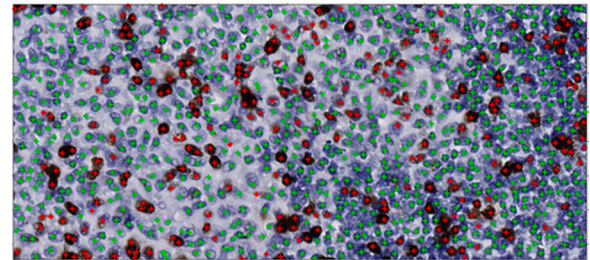
4. Performance evaluation

In this section, the performance of the proposed technique is presented and compared with a few state-of the art techniques for PI calculation, namely FCM + KM (Mungle and Tewary, 2017), GMM + CNN (Saha et al., 2017), Otsu + SVM (Alheejawi et al., 2019a), PI-SegNet (Alheejawi et al., 2019b). As UNet and DeepLab have become very popular CNN architecture, these CNN architectures have also been implemented in this work and the performance has been compared with the proposed technique. Henceforth, these techniques are referred to as PI-UNet and PI-DeepLab. In these techniques, the nuclei segmentation is done using UNet (Ronneberger et al., 2015) and DeepLabV3 (Chen et al., 2018), and the PI calculation is done using the algorithm presented in Section 3.5. In the following, the nuclei segmentation performance is first evaluated and compared. The PI calculation performance is presented next.

Optimal parameters were used for performance evaluation of all techniques. In FCM + KM technique, two clusters (nuclei and background) were used for the FCM and two classes (active and passive) were used for the KM algorithm. In Otsu + SVM, one optimal threshold was used for nuclei and background segmentation, and Gaussian kernel was used for the SVM. The layer configurations of the CNN architectures in the PI-SegNet, PI-UNet, PI-DeepLab, and the proposed technique are



(a) Input Image



(b) Detection Results

Fig. 12. Magnified window with the highest PI value ($PI = 24.14$) in Fig. 11. (a) input image (b) Detection results with 252 active and 792 passive nuclei. The ground truth $PI = 25.15$ with 252 active and 750 passive nuclei.

shown in Table 3. It is observed that the number of layers varies between 37–100 in the various CNNs. Note that the GMM + CNN is not included in the segmentation performance comparison as the technique does not perform nuclei segmentation.

Fig. 13(c)–(h) show visual examples (for subjective assessments) of the segmentation results of FCM + KM, Otsu + SVM, PI-SegNet, PI-UNet, PI-DeepLab and the proposed technique, respectively. In Fig. 13(c)–(g), the passive and the active nuclei are shown with blue and green contours, respectively. A few coarse nuclei boundaries can be observed in

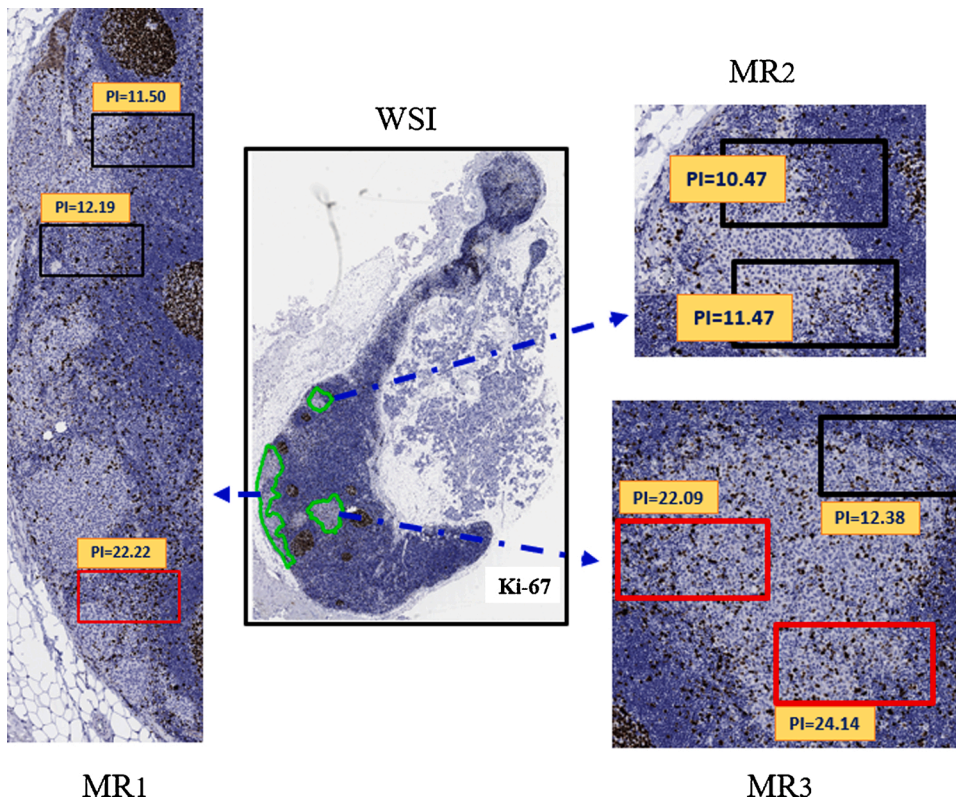


Fig. 11. Example of PI calculation of ROIs in the three melanoma regions (MR_1 , MR_2 , MR_3) in a WSI. The three melanoma regions in the WSI are shown with magnification. The regions MR_1 and MR_2 contains three blocks (with red contours) with the highest PI values: 24.14, 22.22, and 22.09. The other five blocks (with black contours) are also active areas, but the PI values are smaller. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

Table 3

Layer configuration of the deep learning techniques.

Technique	No. of layers in CNN	Details of CNN architecture
PI-SegNet (Alheejawi et al., 2019b)	37	1-In,9-Conv, 8-Btch, 8-Rlu, 4-Mx, 4-Up, 1-Dth, 1-Sft,1-Px
PI-UNet	58	1-In, 19-Conv, 22-Rlu, 4-Mx, 4-Up, 2-Dr, 4-Dth,1-Sft,1-Px
PI-DeepLab	100	1-In,29-Conv, 28-Btch, 25-Rlu, 2-Cr, 1-Mx, 2-Up, 8-Ad, 2-Dth, 1-Sft,1-Px
Proposed Technique	67	1-In, 19-Conv, 18-Rlu, 18-Btch, 4-Mx, 4-Up, 1-Dth, 1-Sft,1-Px

Notations	
In: Input image normalization Layer	Conv: Convolutional Layer
Mx: Max pooling layer	Btch: Batch normalization Layer
Up: Up-sampling layer	Rlu: ReLU Layer
Dr: Dropout layer	Dth: Depth concatenate layer
Px: Pixel classification layer	Sft: Softmax layer
Ad: Addition layer	Cr: Crop layer

Fig. 13(c) and (d). In addition, a part of an active nuclei may be segmented as passive nuclei. This is because these techniques (FCM, Otsu) are based on thresholding and do not consider the shape features of the nuclei during the segmentation process. On the other hand, the PI-SegNet, PI-UNet, and PI-DeepLab techniques segment the nuclei with respect to their morphological features (see Fig. 13(e)–(g)), and results in higher accuracy. Note that the PI-SegNet, PI-UNet, and PI-DeepLab architectures include several pooling layers. As the pooling operation typically causes loss of morphological features (e.g., shape feature) of nuclei, it is likely to degrade the quality of the nuclei segmentation results. In addition, the PI-SegNet and PI-UNet architectures have a large number of convolutional layers at the encoder/decoder side, which may cause vanishing gradients problem. Fig. 13(g) shows the output obtained by the proposed technique, and it is observed that the proposed architecture provides an excellent nuclei segmentation performance.

Table 4 compares the segmentation performance of FCM + KM, Otsu + SVM, PI-UNet, PI-SegNet and PI-DeepLab techniques with the proposed technique using the testing dataset (see section 2). As observed in Table 4, the proposed technique provides the best performance with an

average 94 % accuracy, 89 % IoU and 82 % BF score for five-fold cross validations. Note that the training and the testing dataset are kept completely independent to ensure an unbiased performance comparison.

The PI calculation performance of the proposed technique is now evaluated. The Mean Absolute Error (MAE), and the Mean Percentage Error (MPE) are used as the performance measures, which are defined as follows:

$$MAE = \frac{1}{K} \sum_{k=1}^K |G(k) - B(k)| \quad (12)$$

$$MPE = \frac{1}{K} \sum_{k=1}^K \left| \frac{G(k) - B(k)}{G(k)} \right| \times 100\% \quad (13)$$

where $G(k)$ and $B(k)$ are respectively the ground truth and the obtained PI values for the k th test image, and K is the number of images.

The PI calculation performance of the proposed technique is compared with the state-of-the-art techniques in Table 5. For each testing set, the MAE and MPE values are calculated using Eqs. 12 and 13 for 14 ROI windows. With five-fold cross-validation, 5 sets of MAE and MPE values are obtained, and the average MAE and MPE values are shown in Table 5. It is observed that the FCM + KM technique calculates PI values with highest error (i.e., MAE and MPE values). This is perhaps because, in the FCM + KM technique, the active and the passive nuclei pixels are separated using the K-Means algorithm, which may have low accuracy due to the intensity variation of the pixels. The GMM + CNN

Table 4

Segmentation performance of the proposed technique.

Technique:	Accuracy	IoU	BF Score
FCM + KM (Mungle and Tewary, 2017)	0.74	0.73	0.72
Otsu + SVM (Alheejawi et al., 2019a)	0.78	0.76	0.78
PI-SegNet (Alheejawi et al., 2019b)	0.86	0.80	0.70
PI-UNet	0.84	0.75	0.71
PI-DeepLab	0.88	0.82	0.71
Proposed Technique	0.94	0.89	0.82

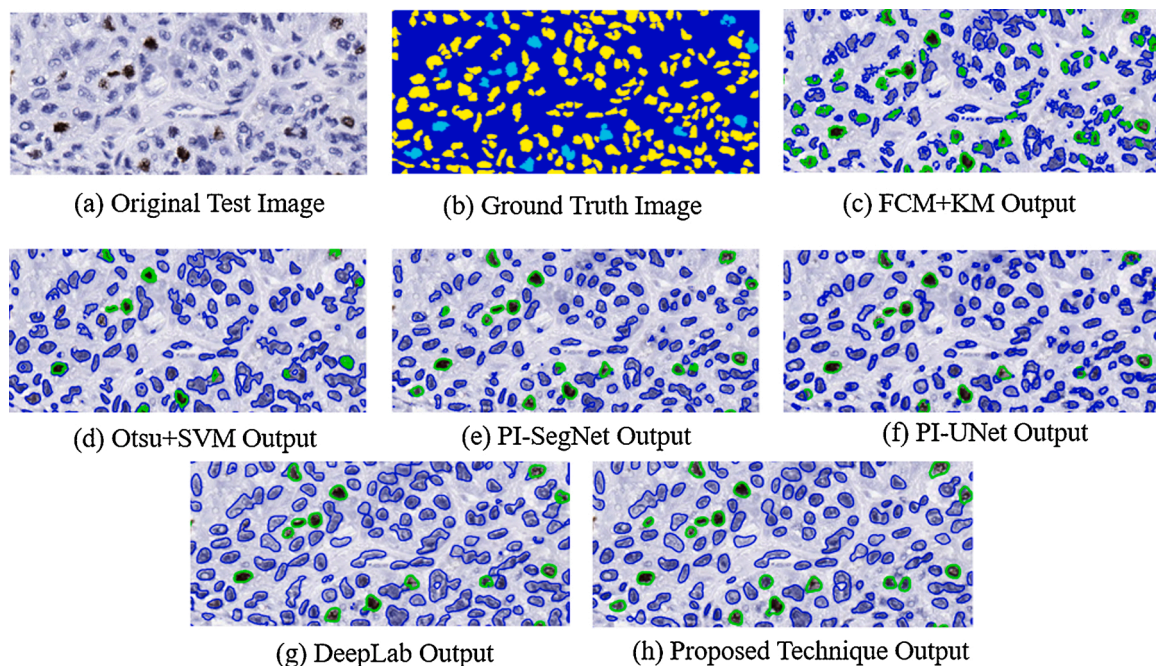


Fig. 13. Subjective comparison of segmentation results (a) original test image, (b) The ground truth image (c)–(h) Segmentation results for FCM + KM, Otsu + SVM, PI-SegNet, PI-UNet, PI-DeepLab and the proposed technique, respectively.

Table 5
Comparison of PI calculation performance.

Technique	MAE	MPE	Execution time (in s)
FCM + KM (Mungle and Tewary, 2017)	3.76	31.34	90
Otsu + SVM (Alheejawi et al., 2019a)	0.89	7.37	16
GMM + CNN (Saha et al., 2017)	2.45	18.89	130
PI-SegNet (Alheejawi et al., 2019b)	0.68	5.35	14
PI-UNet	0.71	5.55	16
PI-DeepLab	0.61	4.59	20
Proposed Technique	0.52	3.92	18

technique provides performance better than FCM + KM, although the RMSE and MAE values are still high. This is perhaps because this technique does not perform nuclei segmentation. Instead, it generates patches around the detected nuclei seeds and classifies the patches as active or passive. The CNN classifier cannot reliably classify a patch that contains both the active and the passive nuclei, resulting in a lower performance.

The PI-SegNet and PI-UNet provide a better and somewhat comparable performance. The PI-DeepLab provides improved results compared to SegNet and UNet. The proposed technique provides the best performance (i.e., lowest error rate).

The computational complexity of various CNN-based techniques, in terms of execution time, is now presented. All techniques were implemented using MATLAB R2020a, and CNNs were implemented using the Deep Network Designer Toolbox. All experiments are performed on a Windows 10 computer with Intel i7-4790 CPU, 12 GB RAM, and NVIDIA GeForce GTX 745 graphic card. The average execution time (for both segmentation and PI calculation) for a WSI in testing mode is shown in the last column of Table 5. It is observed that the GMM + CNN requires the highest execution time. Note that in this technique the GMM (for nuclei detection) takes about 90 % of the overall execution time. FCM + KM technique also requires long execution time due to the FCM and KM algorithms. The other techniques use trained models, are fast in testing mode, and the execution times are comparable. It is observed that the proposed technique has an average of execution time of about 18 s per image.

Table 6 compares the training time required by the CNN architectures (GMM + CNN, PI-SegNet, PI-UNet, PI-DeepLab and the proposed architecture). All four CNN architectures are trained using an NVIDIA GeForce GTX 745 graphic card. Columns 2 and 3 show the training time and the required number of training epochs. The proposed architecture typically finishes the training at around the 7th epoch, and the overall training time is comparable with other CNN architectures.

5. Conclusions

This paper proposes an automated technique for proliferative index calculation in lymph node histopathological images. The proposed technique first determines the melanoma regions in a Ki-67 image by registering it with a corresponding MART-1 image. The technique then uses a CNN architecture to segment and classify the pixels in the melanoma regions of the Ki-67 stained image into three classes (i.e., background, active and passive nuclei). The PI values are then calculated based on the number of the active and passive nuclei. Experimental results show that the proposed technique provides an excellent performance in terms of segmentation (with 94 % segmentation accuracy) and PI calculation results (with 4 % mean error).

CRedit authorship contribution statement

Salah Alheejawi: Methodology, Software, Investigation, Validation, Writing - original draft. **Richard Berendt:** Conceptualization, Data curation, Supervision. **Naresh Jha:** Conceptualization, Data curation, Supervision. **Santi P. Maity:** Investigation, Software, Writing - review & editing. **Mrinal Mandal:** Conceptualization, Supervision, Writing -

Table 6
Training time of CNN models.

Technique	Training time (in M)	Training epochs	No. of Training Parameters
GMM + CNN (Saha et al., 2017)	129	7	189,341
PI-SegNet (Alheejawi et al., 2019b)	141	14	225,987
PI-UNet	180	16	29,379,955
PI-DeepLab	175	6	20,661,638
Proposed Technique	155	7	7,513,247

review & editing, Funding acquisition, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) (Grant number RGPIN-2020-05873). We also acknowledge a scholarship support from the Ministry of Higher Education and Scientific Research (MOHESR) (Iraq).

References

- Akakin, H., Kong, H., Elkins, C., Hemminger, J., et al., 2012. Automated detection of cells from immunohistochemically-stained tissues: application to ki-67 nuclei staining. In: Proc. SPIE 8315, Medical Imaging 2012: Computer-Aided Diagnosis. San Diego, California, USA.
- Al-Lahham, H.Z., Alomari, R.S., Hiary, H., Chaudhary, V., 2012. Automating proliferation rate estimation from ki-67 histology images. In: Proc. SPIE 8315, Medical Imaging 2012, Computer Aided Diagnosis. San Diego, California, USA.
- Alheejawi, S., Xu, H., Berendt, R., Jha, N., Mandal, M., 2019a. Novel lymph node segmentation and proliferation index measurement for skin melanoma biopsy images. *Comput. Med. Imaging Graph.* 73, 19–29.
- Alheejawi, S., Berendt, R., Jha, N., Mandal, M., 2019b. Automated melanoma staging in lymph node biopsy image using deep learning. In: 32nd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). Edmonton, Canada, pp. 6–8. May.
- Bánkfalvi, A., 2000. Comparative methodological analysis of erbB-2/HER-2 gene dosage, chromosomal copy number and protein overexpression in breast carcinoma tissues for diagnostic use. *Histopathology* 37 (5), 411–419.
- Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (12), 2481–2495.
- Chen, L.-C., et al., 2018. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (4), 834–848.
- Csurka, G., Larlus, D., Perronnin, F., 2013. What is a good evaluation measure for semantic segmentation?. In: British Machine Vision Conference. Bristol, UK, pp. 32.1–32.11. Sept 9-13.
- Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. In: 14th International Conference on Artificial Intelligence and Statistics. Ft. Lauderdale, Florida, USA, pp. 315–323. April 11-13.
- Grala, B., Markiewicz, T., Kozłowski, W., Osowski, S., Stodkowska, J., Papier, W., 2009. New automated image analysis method for the assessment of ki-67 labeling index in meningiomas. *Folia Histochem. Cytochem. (Krakow)* 47, 587–592.
- Hanahan, D., Weinberg, R., 2011. Hallmarks of cancer: the next generation. *Cell* 144 (5), 646–674.
- Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Lille, France 32nd International Conference on Machine Learning (ICML), 37, pp. 448–456. July 6–11.
- Kaiming, H., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, USA, pp. 770–778. June 27–30.
- Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Image net classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS). Lake Tahoe, Nevada, USA, pp. 1097–1105. Dec 3–6.
- Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: 2015 Conference on Computer Vision and Pattern Recognition (CVPR). Boston, pp. 3431–3440. June 7–12.
- Mungle, T., Tewary, S., et al., 2017. Automated characterization and counting of Ki-67 protein for breast cancer prognosis: a quantitative immunohistochemistry approach. *Comput. Methods Programs Biomed.* 139, 149–161.

- Murphy, Kevin P., 2012. *Machine Learning, a Probabilistic Perspective*. MIT Press.
- Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: 27th International Conference on Machine Learning (ICML). Haifa, Israel, pp. 807–814. June 21–24.
- Nielsen, P., Riber-Hansen, R., Jensen, T., 2013. Proliferation indices of phosphohistone H3 and Ki67: strong prognostic markers in a consecutive cohort with stage I/II melanoma. *Mod. Pathol.* 26, 404–413.
- Reid, M., Bagci, P., Ohike, N., Saka, B., et al., 2015. Calculation of the Ki67 index in pancreatic neuroendocrine tumors: a comparative analysis of four counting methodologies. *Mod. Pathol.* 28, 686–694, 2015.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: convolutional networks for biomedical image segmentation. In: Munich, Germany International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 9351, pp. 234–241. Oct 5–9.
- Saha, M., Chakraborty, C., Arun, I., Ahmed, R., Chatterjee, S., 2017. An advanced deep learning approach for Ki-67 stained hotspot detection and proliferation rate scoring for prognostic evaluation of breast cancer. *Sci. Rep.* 7.
- Shijie, J., Ping, W., Peiyi, J., Siping, H., 2017. Research on data augmentation for image classification based on convolution neural networks. In: 2017 Chinese Automation Congress (CAC). Jinan, China, pp. 4165–4170. Oct 20–22.
- Siegel, R.L., Miller, K.D., Jemal, A., 2020. Cancer statistics, 2020. *CA Cancer J. Clin.* 70 (January/February), 7–30, 2020.
- Wang, Y., Crookes, D., Eldin, O., Wang, S., Hamilton, P., Diamond, J., 2009. Assisted diagnosis of cervical intraepithelial neoplasia (cin). *IEEE J. Selected Topics Signal Process* 3 (1), 112–121.
- Xu, H., Lu, C., Jha, N., Mandal, M., 2014. An efficient technique for nuclei segmentation based on ellipse descriptor analysis and improved seed detection algorithm. *IEEE J. Biomed. Health Inform.* 18 (5), 1729–1741.
- Xu, H., Lu, C., Berendt, R., Jha, N., Mandal, M., 2017. Automatic nuclei detection based on generalized Laplacian of Gaussian filters. *IEEE J. Biomed. Health Inform.* 21 (3), 826–837.
- Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional neural networks. In: European Conference on Computer Vision (ECCV), 1, pp. 818–833.