Neural Graph Refinement for Robust Recognition of Nuclei Communities in Histopathological Landscape

Taimur Hassan, *Member, IEEE*, Zhu Li, *Senior Member, IEEE*, Sajid Javed, Jorge Dias, *Senior Member, IEEE*, and Naoufel Werghi, *Senior Member, IEEE*

***Abstract*— Accurate classification of nuclei communities is an important step towards timely treating the cancer spread. Graph theory provides an elegant way to represent and analyze nuclei communities within the histopathological landscape in order to perform tissue phenotyping and tumor profiling tasks. Many researchers have worked on recognizing nuclei regions within the histology images in order to grade cancerous progression. However, due to the high structural similarities between nuclei communities, defining a model that can accurately differentiate between nuclei pathological patterns still needs to be solved. To surmount this challenge, we present a novel approach, dubbed neural graph refinement, that enhances the capabilities of existing models to perform nuclei recognition tasks by employing graph representational learning and broadcasting processes. Based on the physical interaction of the nuclei, we first construct a fully connected graph in which nodes represent nuclei and adjacent nodes are connected to each other via an undirected edge. For each edge and node pair, appearance and geometric features are computed and are then utilized for generating the neural graph embeddings. These embeddings are used for diffusing contextual information to the neighboring nodes, all along a path traversing the whole graph to infer global information over an entire nuclei network and predict pathologically meaningful communities. Through rigorous evaluation of the proposed scheme across four public datasets, we showcase that learning such communities through neural graph refinement produces better results that outperform state-of-the-art methods.**

***Index Terms*— Nuclei communities, graph representational learning, histology images, colorectal cancer.**

I. Introduction

WHOLE slide imagery (WSI) is becoming the gold standard for monitoring cancerous growths as it allows pathologists to objectively assess the full extent of malignancy within the cellular tissues at an early stage [1]. Similarly, the increased use of histology scans in computational pathology enabled the researchers to develop automated tools that can assist pathologists in analyzing the tumorous profiles across multi-resolution histopathological landscapes [2]. Nevertheless, storing or transferring such large-sized slides poses a significant challenge to clinicians in their daily routines. In addition, analyzing such multi-gigapixel histology scans as a whole is a practically infeasible and computationally inefficient process [3]. Therefore, the clinicians typically divide the histology slides into non-overlapping patches for the given magnification level in order to overcome the storage and computational issues while objectively visualizing the cancerous spread [4].

The accurate extraction of nuclei communities plays a critical role [5] towards effectively evaluating the cancerous progression within cellular tissues. A nucleus is a middle portion of the cell controlling the genetic information, cell division, protein synthesis, and cellular growth [6]. Moreover, a community is defined as a group or cluster of nuclei that have the same type. Apart from this, communities, representing different types of nuclei, form a community network within the histopathological landscape [7]. Many researchers have proposed sophisticated architectures to detect these communities in order to perform tissue phenotyping [5] and cancerous prognosis [3] tasks. The recent wave of methods also explored few-shot learning and incremental learning paradigms to achieve good detection performance at the inference stage while training the models with limited data [8]. However, to the best of our knowledge, either these methods cannot be scaled to the clinical settings due to architectural design constraints [5], or they are vulnerable in extracting the granularity of the tissue structures depicted within the histology images [4]. Graph-based methods have also been widely explored for recognizing the nuclei communities represented within the histology scans. The performance of these methods is relatively better than the other approaches [9]. However, these frameworks still face difficulty in distinguishing the close contextual and textural appearances of nuclei communities within the WSI landscape [6], [10].

To address the above-mentioned limitations, this paper proposes a neural graph refinement scheme, as shown in Figure 1, that forms a nuclei community network via a fully connected graph in which each node represents a nucleus, and adjacent nodes are connected to each other via an undirected edge. Through iterative refinement across a whole path traversing the graph, contextual information is diffused, and edges are updated via learned schemes until convergence.

Manuscript received 9 March 2023; revised 30 September 2023; accepted 12 November 2023. Date of publication 8 December 2023; date of current version 15 December 2023. This work is supported by a research fund from Advanced Technology Research Center Program (ASPIRE), UAE, Ref: AARE20-279, Khalifa University, UAE, Ref: CIRA-2021-052, the Terry Fox Foundation, Canada, Ref: I1037, and Mubadala, UAE, Ref: MuACPA 2023/2024. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ming Shao. *(Corresponding author: Taimur Hassan.)*

Taimur Hassan is with the Department of Electrical, Computer, and Biomedical Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates (e-mail: taimur.hassan@adu.ac.ae).

Zhu Li is with the Department of Computer Science and Electrical Engineering, University of Missouri–Kansas City (UMKC), Kansas, MO 64110 USA (e-mail: lizhu@umkc.edu).

Sajid Javed and Jorge Dias are with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: sajid.javed@ku.ac.ae; jorge.dias@ku.ac.ae).

Naoufel Werghi is with the Center for Secure Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: naoufel.werghi@ku.ac.ae).

Digital Object Identifier 10.1109/TIP.2023.3337666

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

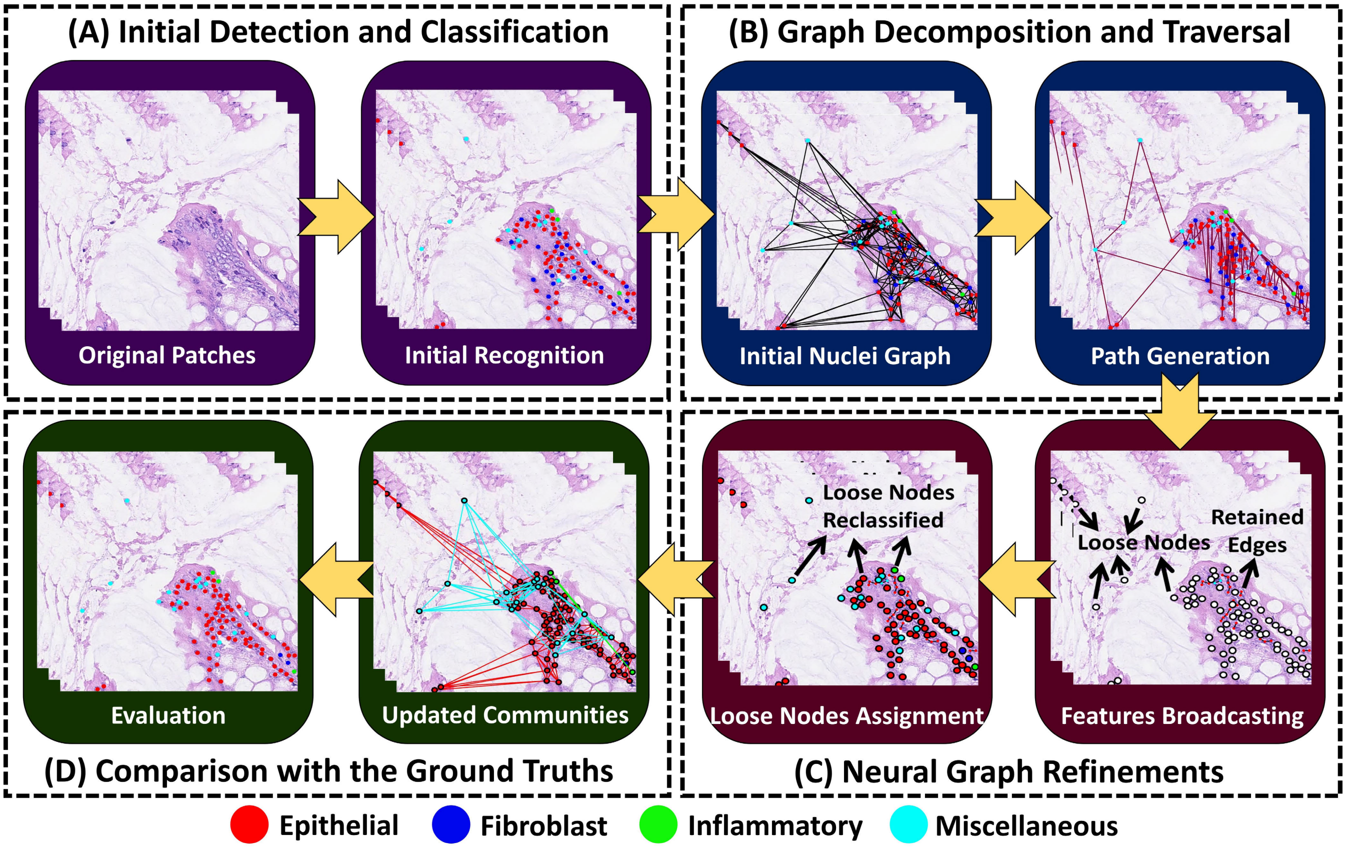

Fig. 1. Schematic illustration of the proposed framework. At first, we perform initial recognition of nuclei within the WSI patches via a pre-trained detection/segmentation model (A). Afterward, we generate a fully connected community graph in which each nucleus is represented as a node and is connected with other nuclei through an undirected edge. To traverse the graph for nuclei label refinement, we generate a Hamiltonian path from the community graph in a breadth-first search manner (B). For each node within the path, we generate the node and edge feature embeddings through neural networks and broadcast these embeddings within the whole graph for a fixed number of iterations. After propagating the embeddings in each iteration, we also utilize their representations to classify each edge (connecting the two adjacent nodes within the graph) as active or non-active. This edge classification retains connections between similar nodes (nuclei) while removing the links between heterogeneous nodes. This edge refinement also leads to the production of loose (disconnected) nodes within the graph. These loose nodes are then subsequently reclassified by exploiting their latent homogeneity (C). Finally, the reclassified nodes aid in producing the updated community graph, which is compared with the ground truth labels (D).

The proposed scheme can be coupled with any detection or segmentation model to drastically improve its performance toward recognizing the heterogeneous and homogeneous nuclei communities within the histopathological landscape. Similarly, the featurization block of the proposed scheme can be integrated with any edge classification, graph partitioning and clustering method to retain only the purest homogeneous) nuclei within the graph, whereas the rest of the connections between heterogeneous nodes (belonging to different nuclei categories) are eliminated. The integration of proposed scheme with the graph partitioning and clustering methods also opens a door for the proposed scheme to be applied to a vast variety of applications involving graph processing and graph representational learning. Apart from this, the extent of the proposed system is tested across four public benchmark datasets containing a wide variety of nuclei structures. The proposed framework has outperformed state-of-the-art methods toward accurately recognizing the nuclei communities in a diversified histopathological landscape across all these four datasets.

II. Related Works

Many researchers have worked on extracting nuclei communities using histology images. In the following, we present a brief overview of the main approaches. We refer the readers to [11] for a detailed survey.

A. Classical Methods

Initial methods designed for monitoring cancerous pathologies were based on handcrafted features related to the shape, texture, and spatial characteristics of the nuclei communities depicted within the WSI patches [12]. For example, Nguyen et al. [13] analyzed the shape features to differentiate benign and malignant nuclei within prostate WSI patches. Yuan et al. [14] extracted the lymphocytes and stroma tissues from the histology scans by computing the morphological features. Although classical machine learning methods were widely used for diagnosing cancerous pathologies, the robustness of these methods was confined to minimal experimental settings due to the subjectiveness of the extracted features. Therefore, when exposed to a wider variety of tissues within WSIs, their performance deteriorated drastically [15].

B. Deep Learning Methods

Deep learning has significantly improved cancer progression monitoring using histology images by enhancing the extraction and the classification of the nuclei communities [16] and cellular tissue pathologies [3]. For example, Kim et al. [17] used a pre-trained segmentation model along with principal component analysis and discrete wavelet transform to segment cellular pathologies from the patched histology images.

Graham et al. [5] developed Hover-Net [5], which contains residual blocks and distance encoders to perform simultaneous segmentation and classification of nuclei communities from the WSI patched sequences. Similarly, Tripathi and Singh [18] used a conjunction of CNN features that are fed to the multilayered perceptron to recognize nuclei types. Unsupervised classification of cell communities has also been attempted. Representative works are reported in [19].

C. Graph-Based Methods

Despite advancing the state-of-the-art in computational pathology, deep learning methods are still vulnerable to high correlation within textural and contextual representations of the whole slide imagery, which limit their capacity to discriminate different grades of cancerous tissues. To address these challenges, researchers investigated the utilization of graph-based approaches in cancer image analysis [9], [20]. For example, Ramirez et al. [20] employed graph convolutional networks to classify different types of cancerous pathologies. Lu et al. [6] proposed SlideGraph$^+$ to predict human epidermal growth factor receptor-2 (HER2) from the breast cancer H&E stained whole slide images. SlideGraph$^+$ represents the multi-gigapixel WSI into an undirected graph by first clustering the WSI patches that depict similar tissue structures using the Delauney triangulation scheme [21]. The constructed graph is then passed to the custom graph neural network architecture, which extracts distinct feature representation via edge convolutional layers to predict the presence of HER2 patterns.

Although these recent frameworks evidenced the importance of the network representation as reflected in the superior performance compared to their predecessors. There is still a significant performance gap to bridge toward reaching a thoroughly reliable system.

III. Contributions

We present a novel neural graph refinement (NGR) scheme that iteratively exploits the homogeneity within the cellular tissue pathologies in order to robustly extract the nuclei communities within the whole slide images. The accurate recognition of the nuclei communities also enables screening and grading of the colorectal cancer severity and facilitates the pathologists towards timely mitigating cancer spread by giving effective prognosis. More specifically, the main contributions of this work are:

- This paper presents a novel, ordered, continuous, and iterative neural graph refinement scheme that formulates a whole path to traverse a complete community graph in order to diffuse cellular contextual information within colorectal cancer whole slide imagery for improving the nuclei community recognition.
- In addition to this, the proposed NGR scheme can be coupled with any clustering and graph partitioning scheme to separate homogeneous and heterogeneous nuclei types by analyzing their associations within the latent space (see Sections VI-A.7 and VI-A.8 for more details).

We also want to highlight that in our previous work [10], we pioneered investigating a message-passing network for cell classification in the whole slide images. In that preliminary investigation, we constructed the nearest neighbor graph such that the nodes represent the nuclei centroids, and the adjacent nodes are formed through K nearest neighbor connectivity. Moreover, for each node and the set edges connecting it with the neighboring nodes, we compute distinct feature messages that transmit the contextual information to the adjacent nodes while updating the nodes' connections.

In this paper, we present a more robust and structured solution that exploits the homogeneity between the nuclei community by coupling neural featurization process with edge classification, graph partitioning and clustering schemes in order to well-discriminate similar textured nucleus types within the histopathological landscape. The main distinctions between the proposed framework with our previous work [10] are: 1) We employ a fully connected graph of the nuclei community whereby we explicitly connect each node with every other node based on the vicinity in the space of features, whereas in our previous work, we used a graph constructed with the nearest neighbor technique in the spatial domain. With the so-constructed graph, we enhance the pruning and the retaining of the edge between, respectively, heterogeneous nodes (belonging to different communities) and homogeneous nodes (belonging to the same community). 2) Rather than transferring node and edge features in the form of messages to selected node pairs in a K-nearest neighbor graph, we developed an iterative broadcasting scheme propagating node and edge features across the whole graph. Apart from edge classification, we also integrated the featurization process of the proposed scheme with traditional graph partitioning and clustering methods for performing the nucleus classification tasks. See Sections VI-A.7 and VI-A.8 for more details. 3) The contextual information is diffused to the neighboring nodes across a Hamiltonian path traversing the whole graph, ensuring thus ordered analysis and neighborhood continuity. In our previous work, the nodes were browsed in an arbitrary fashion according to their arrangement in the nearest neighbor graph structure. 4) Consistently with the Hamiltonian path concept, we propose a novel scheme in which the adjacent nodes are decomposed into downstream nodes (that receive embedding information) and upstream nodes (that send embedding information) depending on their relative positions in the traversal path with respect to the current node. 5) The node and edge features are updated across each broadcasting iteration through seven learnable multilayered perceptrons (MLPs) rather than non-trainable message-passing networks [10]. 6) We propose a new classification stage, where we re-classify the loose nodes generated during the graph refinements by analyzing their latent space homogeneity with each nuclei cluster.

The rest of this paper is organized as follows: Section IV explains the proposed scheme in detail. Section V presents the details about the experimental protocol, while Section VI showcases the detailed validation of the proposed framework through various experiments and its comparison with state-of-the-art methods using different metrics. Section VII concludes the manuscript while highlighting the future directions.

IV. Proposed Method

The proposed scheme is divided into four phases, as shown in Figure 1. In the first phase, the initial recognition of the nuclei communities is performed using the pre-trained detection or segmentation models. In the second phase, the centroids of the detected cells are utilized in generating a fully connected graph of the nuclei communities from which we derive a Hamiltonian path that browses the whole graph. This traversal path allows us to operate on each node within the community to extract its appearance and geometrical features. In the third

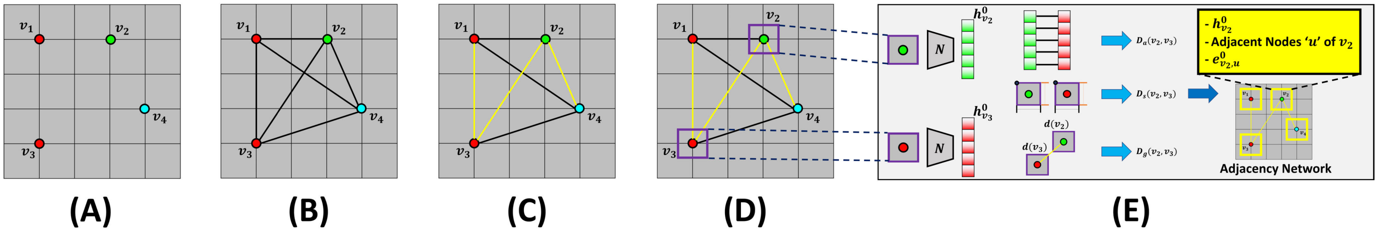

Fig. 2. Working schematic of the initial feature extraction pipeline. (A) Initially detected nuclei. (B) Construction of the fully connected community graph. (C) Generation of the traversal path $\mathbb{P}$ (in yellow) starting at v_1. (D) Establishing neighborhood a_v, in the form of a bounding box around the nucleus (e.g., the neighborhoods a_{v_2} and a_{v_3} of v_2 and v_3). (E) Computation of initial node feature vectors from the neighborhood a_v via deep feature extractor N (e.g., the $h^o_{v_2}$ and $h^o_{v_3}$). From the neighborhoods a_v and the initial feature vectors, we compute the edge descriptor $e^0_{v_2,v_3}$ aggregating the appearance edge features (D_a), spatial edge features (D_s) and geodesic edge features (D_g). The traversal path, the neighborhoods, the node feature vectors, and the edge descriptor are then used to construct and initialize the adjacency network $\mathbb{A}$.

phase, appearance and geometrical features are updated and propagated across the graph for a fixed number of iterations. In each iteration, we also classify each link, connecting two adjacent nodes, using node and edge features to enforce maintaining (respectively eradicating) links between homogeneous (respectively heterogeneous) nodes. This procedure also generates loose nodes (i.e., nodes having no connection with any other nodes) in the updated community graph. We assign these loose nodes to their corresponding community by searching their closest (homogeneous) nuclei cluster within the latent feature space. Lastly, in the fourth phase, we compare the refined communities in the WSI patches with their ground truths to evaluate the degree of improvements achieved by the proposed scheme as compared to the initial results. While the last phase is actually an assessment phase and cannot be qualified as part of our detection and classification scheme, we put it in the pipeline because it employs a non-classic metric for the evaluation. A detailed description of all four phases is presented in the subsequent sections.

A. Initial Cell Detection and Classification

In the first stage, we perform the initial detection and classification of nuclei communities using conventional detection and segmentation models, such as Faster R-CNN [22], Mask R-CNN [23], and Hover-Net [5]. The detection models recognize and localize the nuclei communities through bounding boxes, while the segmentation models localize them through masks. We also derive the bounding boxes from these masks by analyzing the minimum and maximum extent of each connected component representing each nuclei type. These initial nuclei recognition contain a large number of misclassifications since these models employ region-based [22], [23], and edge-based [5] searches, which are susceptible to a high contextual and textural correlation between cellular tissue structures [5], [8], as evident from Figure 1(A).

B. Graph Decomposition and Traversal

In this stage, we construct a fully connected (complete) undirected graph from the initially recognized nuclei within the WSI patches. We also derive the traversal path and compute the initial node and edge feature vectors. Afterward, we encapsulated all these entities into a structure that we dubbed the Adjacency Network. Figure 2 illustrates the proposed scheme. More details are presented in the subsequent sections.

1) Complete Graph Construction: We initially construct a fully connected graph $\mathbb{G}(V, E)$, where V is the set of the nodes representing the detected nuclei and E are the related edges (see Figure 1-B, and Figure 2-B). A fully connected graph is a complete graph in which every pair of distinct nodes is connected by a unique edge. Each node v inherits the bounding box of the associated nuclei, which we refer to as the node neighborhood a_v. This neighborhood is defined in terms of coordinates of the left upper corner of the bounding box (x_v, y_u), its width w_v, and height h_v.

2) Path Generation: After constructing $\mathbb{G}$, we derive from it a Hamiltonian path $\mathbb{P}$, starting at any arbitrary node v_s and traversing the whole graph without passing through the given node v more than once (see Figure 2-C). We construct $\mathbb{P}$ in a simple breadth-first manner (see Algorithm-1). $\mathbb{P}$ allows us to define the concept of downstream and upstream nodes with respect to the current node v. The downstream nodes are the nodes in $\mathbb{P}$, adjacent to v, that come before v in the path. The upstream nodes are the adjacent nodes to v in $\mathbb{P}$, which come after v. For instance, in Figure 2 (C), where $v_s = v_1$, the downstream and upstream nodes for v_3 are v_1 and (v_2, v_4), respectively. The path $\mathbb{P}$ also allows us to propagate the information about the node features across the whole graph $\mathbb{G}$ in a continuous and ordered fashion. More details will be described in Section IV-C.

3) Adjacency Network Initialization: To extract the node and edge features and elegantly store them, we introduce the concept of an adjacency network. An adjacency network, $\mathbb{A}$, is a structure that encompasses the traversing path $\mathbb{P}$ augmented at each node v with both node's and edge's descriptors. The node descriptor is the initial feature vector (h^o_v) derived from the node neighborhood a_v via a deep feature extractor N. The N model and its training are described in Section IV-E. The edge descriptor $e_{v,u}$ is a feature vector describing the connectivity between the pair of nodes (v, u). This vector $e_{v,u}$ is an aggregation of appearance edge features (D_a), spatial edge features (D_s), and geodesic edge features (D_g) (i.e., $e_{v,u} = [D_a, D_s, D_g]$). The visualization of h^o_v and $e_{v,u}$ are shown in Figure 2 (E). The detailed descriptions of D_a, D_s, and D_g are presented below:

a) Appearance edge features: D_a evaluates the similarity appearance between a pair of neighborhoods, which we define in terms of the Euclidean distance between the node feature vectors (h^o_v) and (h^o_u).

$$D_a = ||h^o_v - h^o_u||. \tag{1}$$

b) Spatial edge features: D_s measures the spatial closeness of the neighborhoods associated with the two nodes v

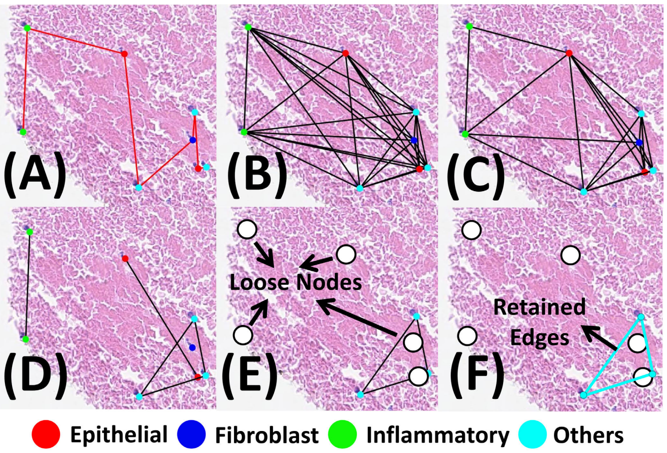

Fig. 3. Graphical illustration of the proposed neural graph refinement (NGR) scheme. (A) Generated path $\mathbb{P}$, (B) fully connected community graph, (C) showcases the outcome of the first NGR iteration (notice the pruned edges), (D) shows the outcome of the second NGR iteration (notice the further pruned edges). (E) shows the final NGR iteration where loose nodes are highlighted. (F) shows the retained nuclei community.

Algorithm 1 Adjacency Network Initialization

```
Input: Path $\mathbb{P}$, Detected Nuclei $\mathbb{D}$
Output: Adjacency Network $\mathbb{A}$
$\mathbb{A} \leftarrow$ ConstructGraph($\mathbb{D}$)
foreach node $v$ in $\mathbb{P}$ do
    $h_v^o \leftarrow N(a_v)$
    $\mathbb{A} \leftarrow$ Update($\mathbb{A}$,$h_v^o$)
    $\Psi_v \leftarrow$ AdjacentNodes($v$, $\mathbb{A}$)
    foreach node $u$ in $\Psi_v$ do
        $e_{v,u} \leftarrow [D_a, D_s, D_g]$
        $e_{v,u}^0 \leftarrow N_0^e(e_{v,u})$
    end
end
return $\mathbb{A}$
```

and u via their bounding boxes' parameters.

$$D_s = \left[\frac{2(x_u - x_v)}{h_u + h_v}, \frac{2(y_u - y_v)}{w_u + w_v} \right]. \tag{2}$$

c) Geodesic edge features: D_g measures the algebraic geodesic distance between the two nodes in the path $\mathbb{P}$. Here, we approximate this distance by the signed length of the sequence between the two nodes defined as follows:

$$D_g = d\mathbb{P}_u - d\mathbb{P}_v, \tag{3}$$

where $d\mathbb{P}_u$ is the length of the sequence of nodes from the start node v_s in the path $\mathbb{P}$ to the node u, and $d\mathbb{P}_v$ is the length the sequence of nodes from v_s in the path $\mathbb{P}$ to the node v. Also, note that the commutative property does not hold for the edge descriptor, i.e., $e_{v,u} \neq e_{u,v}$. Afterward, $e_{v,u}$ is passed to the initial feature extractor N_e^0 to generate the initial edge descriptor $e_{v,u}^0$. The N_e^0 model and its training are described in Section IV-E. The adjacency network initialization algorithm is depicted in Algorithm-1.

C. Neural Graph Refinement

The proposed neural graph refinement (NGR) is an iterative scheme that analyzes the homogeneity between nodes (i.e., nuclei communities) by propagating their deep feature representations across the graph. At each iteration k, the proposed scheme traverses the path $\mathbb{P}$, performing, at each node v, two tasks: a) updating the edge connection between the node v and its adjacent nodes, and b) updating the node descriptor for the next iteration.

1) Edge Connection Update: The edge descriptor defining the connection between the node v and an adjacent node u is updated using a learned feature extractor model, N_e^p, fusing node and edge descriptors of the previous iterations, as shown below:

$$e_{v,u}^k = N_e^p(h_v^{k-1}, h_u^{k-1}, e_{v,u}^{k-1}). \tag{4}$$

The N_e^p model and its training are described in Section IV-E. Next, the newly updated edge descriptor is passed through the binary classifier N_e^c, which outputs $y_{v,u} \in \{0, 1\}$. $y_{v,u} = 1$ indicates that the edge between node v and u is preserved, and $y_{v,u} = 0$ indicates that the edge between v and u is eliminated due to large heterogeneity between the latent representation of both nodes.

2) Node Vector Update: The node vector update is performed by first partitioning the adjacent nodes into two groups which we dub downstream nodes (Ds) and upstream nodes (Us). The downstream nodes are nodes in the traversal path $\mathbb{P}$ that come before the current node v, whereas the upstream nodes are those which come after v. Considering the algebraic geodesic distance (Eq. 3), the two groups can be differentiated by a negative D_g and a positive D_g for (Ds) and (Us), respectively. For example, in Figure 2 (C), we can observe that for the node v_3, Ds and the Us are composed of v_1, and (v_2, v_4), respectively. The proposed partitioning of nodes into Ds and Us aims to incorporate the complete contextual information into the node descriptors while embedding the sequential geodesic aspect of the path $\mathbb{P}$.

We first update the node descriptors, for a node v, separately for each group by deriving two deep feature vectors $h_{Ds_v}^k$ and $h_{Us_v}^k$ as follows:

$$h_{Ds_v}^k = \sum_{u \in Ds}^{n_{Ds}} N_{Ds}(h_v^0, h_v^{k-1}, e_{v,u}^k), \tag{5}$$

$$h_{Us_v}^k = \sum_{u \in Us}^{n_{Us}} N_{Us}(h_v^0, h_v^{k-1}, e_{v,u}^k), \tag{6}$$

where n_{Ds} denotes the number of downstream nodes, n_{Us} denotes the number of upstream nodes, N_{Ds} and N_{Us} represent the feature extractor models associated to the downstream and the upstream groups, respectively. Here, it should be noted that we use two different models, N_{Ds} and N_{Us}, to generate downstream and upstream node descriptors w.r.t node v, respectively. This is because the edge descriptors $e_{v,u}^k$ and $e_{u,v}^k$ are not necessarily identical. Section IV-E provides more details about these models. The two deep feature vectors are derived by integrating the previous node descriptor h_v^{k-1}, the current edge descriptor $e_{v,u}^k$, and also the initial node descriptor h_v^0. The rationale for including this third component is to keep a certain balance with respect to the initial cell partition.

In the next stage, $h_{Ds_v}^k$ and $h_{Us_v}^k$ are concatenated and are fed to the feature extractor model N_n^p to generate the node vector update for the current iteration k. More details about the models are in Section IV-E.

$$h_v^k = N_n^p([h_{Ds_v}^k, h_{Us_v}^k]). \tag{7}$$

Algorithm 2 Neural Graph Refinement

```
Input: Path $\mathbb{P}$, Adjacency Network $\mathbb{A}$
Output: Updated $\mathbb{A}$
$\mathbb{N} \leftarrow \phi$
$i_t \leftarrow$ max_iteration // propagation iterations
$k \leftarrow 1$
convergence $\leftarrow$ False
while not convergence do
    if $k \geq i_t$ then
        break // max iterations reached
    end
    foreach node $v$ in $\mathbb{P}$ do
        $\Psi_v \leftarrow$ AdjacentNodes($v$, $\mathbb{A}$)
        $h^k_{Ds_v} \leftarrow \phi$
        $h^k_{Us_v} \leftarrow \phi$
        foreach node $u$ in $\Psi_v$ do
            $e^k_{v,u} \leftarrow N^p_e(h^{k-1}_v, h^{k-1}_u, e^{k-1}_{v,u})$
            $y_{v,u} \leftarrow N^c_e(e^k_{v,u})$
            $\mathbb{A} \leftarrow$ Update($\mathbb{A}, v, u, e^k_{v,u}, y_{v,u}$)
            if $d\mathbb{P}_u < d\mathbb{P}_v$ then
                $h^k_{Ds_v} \leftarrow h^k_{Ds_v} + N_{Ds}(h^0_v, h^{k-1}_v, e^k_{v,u})$
            end
            if $d\mathbb{P}_u > d\mathbb{P}_v$ then
                $h^k_{Us_v} \leftarrow h^k_{Us_v} + N_{Us}(h^0_v, h^{k-1}_v, e^k_{v,u})$
            end
        end
        $h^k_v \leftarrow N^p_n([h^k_{Ds_v}, h^k_{Us_v}])$
    end
    convergence $\leftarrow$ IsConverged() // See Eq. (9)
    $k \leftarrow k + 1$
end
return $\mathbb{A}$
```

h^k_v embeds a rich contextual and history exposure allowing the best discrimination of the heterogeneous node pairs (v, u) and eventually breaking the links between them. On the opposite, if the nodes (v, u) are homogeneous, it builds stronger latent space associations that can strengthen the edge features and aid in retaining the connection between them.

The full NGR algorithm is depicted in Algorithm 2. Figure 3 depicts the outcome of each NGR iteration, through which we can analyze how the links between heterogeneous nodes are pruned while the links between homogeneous nodes are retained.

3) Loose Nodes Assignment: The NGR scheme produces subgraphs of connected nodes encompassing the different cell categories and a set of loose nodes having no connections. To re-classify these loose nodes, we utilize their node features, generated through N, to derive the latent features, and then we assign to it the class of the closest node, in the connected cohorts, based on a simple Euclidean distance as depicted in Algorithm 3.

D. Comparison With the Ground Truths

We evaluate the recognition performance of the proposed scheme by comparing the centroids of the recognized nuclei with the ground truths. The performance scores are then compared with the state-of-the-art solutions as discussed further in Section VI.

Algorithm 3 Loose Nodes Assignment

```
Input: Path $\mathbb{P}$, Adjacency Network $\mathbb{A}$
Output: None
min $\leftarrow$ double.max
foreach node $v$ in $\mathbb{P}$ do
    $\Psi_v \leftarrow$ AdjacentNodes($v$, $\mathbb{A}$)
    if $\Psi_v$ is $\phi$ then
        foreach node $u$ in $\mathbb{P}$ except $v$ do
            $h_v \leftarrow N(a_v)$
            $h_u \leftarrow N(a_u)$
            if $\|h_u - h_v\| <$ min then
                min $\leftarrow \|h_u - h_v\|$ // Euclidean Distance
                $v$.Type $\leftarrow$ $u$.Type
            end
        end
    end
end
```

TABLE I

Detailed Description of the Trainable Networks Within the Proposed Framework. The Abbreviations Are: 'FCL': Fully Connected Layer, 'w/o': Without

Model	Input Size	Output Size	Number of Layers
N	224 × 224 × 3	1 × 4096	174 w/o FCL
N^0_e	1 × 4	1 × 16	4 w/o FCL
N^p_e	1 × 4112	1 × 16	4 w/o FCL
N_{Ds}	1 × 4112	1 × 2048	4 w/o FCL
N_{Us}	1 × 4112	1 × 2048	4 w/o FCL
N^p_n	1 × 4096	1 × 2048	4 w/o FCL
N^c_e	1 × 4096	1 × 2	7 with FCL

E. Trainable Feature Extraction Models

In total, we used seven trainable networks in the proposed scheme to extract node and edge features and classify the links between heterogeneous and homogeneous nuclei. Table I reports the detailed architecture of these models. All of these models are trained using categorical cross-entropy loss function, except N^c_e, which is trained using binary cross-entropy loss function. A further description of these models and their training strategy is discussed in the subsequent sections.

1) Initial Node Encoder: The node encoder (N) is responsible for generating the feature embeddings from the detected nuclei. Here, we utilize the bounding boxes of the detected nuclei to crop them out from the WSI patches. Each cropped area is then resized and passed to N to generate initial node features. N is typically a pre-trained classification network that contains a feature extractor and a classification head consisting of fully connected layers. We train this network first to recognize nuclei categories from the cropped nuclei patches extracted from the WSI patches (in the training set) using the provided nuclei annotations. After training, we remove the classification head from N and use it as a node feature extractor.

2) Initial Edge Encoder: The initial edge encoder (N^0_e) is responsible for generating the initial edge descriptor using the appearance and geometric features that are expressed in Eq. 1, 2, and 3. Architecturally, N^0_e is a four-layered multilayer perception (MLP), in which we also append a fully connected layer during training to tune its weights for classifying the nuclei categories within the training WSI patches. The input which it takes to predict these nuclei categories is the edge descriptor $e_{v,u}$ that has the size of 1×4. Moreover, to train N_e^0, we compute $e_{v,u}$ for each nuclei category across the training samples of each dataset and store them on disk to formulate a training set for N_e^0. After tuning the weights of N_e^0 towards recognizing the nuclei categories, we remove the fully connected layer from it and use it as a feature extractor.

3) Edge Update Encoder: The edge update encoder (N_e^p) updates the edge features across the graph network within the NGR scheme. In each iteration k, the input to N_e^p is a concatenated representation of node and edge features from the previous $k-1$ iteration, as expressed in Eq. 4, which produces updated edge features of size 1×4112. This model is a four-layered MLP appended with a fully connected layer at the end for nuclei classification. We train this model with the initial feature vectors h_v^0, h_u^0, and $e_{v,u}^0$ derived from training data. After training, we remove the classification head to use the model as a feature extractor.

4) Node Downstream and Upstream Encoders: The downstream node encoder N_{Ds} is another four-layered MLP that generates the feature embeddings from the downstream nodes using the current k^{th} iteration edge update embeddings from node v to u, i.e., $e_{v,u}^k$, initial node embeddings (generated through N), i.e., h_v^0, and node update embeddings produced in the previous $k-1$ iteration, i.e., h_v^{k-1}. The output embeddings, generated through N_{Ds}, are then utilized in computing the node update descriptor, as expressed in Eq. 7. The N_{Ds} model is devised by appending a fully connected layer to perform nuclei classification tasks, and it is trained using the feature vectors $e_{v,u}^k$ and h_v^0, where u belong to the downstream nodes of v. After training, we remove the fully connected layer and use the trained weights of N_{Ds} to produce a downstream feature descriptor of size 1×2048 from the input samples of size 1×4112. We use an identical scheme for the upstream encoder N_{Us}, except that nodes u are selected from the upstream group of v.

5) Edge Classifier: The edge classifier N_e^c is responsible for classifying each edge within the community graph $\mathbb{G}$ as active or non-active during each NGR iteration. The non-active edges are then pruned from the $\mathbb{G}$, and the active edges are retained across each iteration. N_e^c is a seven-layered architecture, having the input size of 1×4096 and the output size of 1×2. The input sample for N_e^c consists of an updated edge descriptor, generated in each k^{th} iteration, i.e., $e_{v,u}^k$. Similarly, to train the N_e^c, the edge update embeddings are derived from the training WSI patches of each dataset for offline training. The training set is composed of $e_{v,u}$ samples and their corresponding state (active or non-active) as derived from the nuclei communities in the annotated datasets.

6) Node Update Encoder: The node update encoder (N_n^p) is utilized in the NGR scheme to update the feature representations across each node in the k^{th} iteration. The input to N_n^p is a combined feature representation of N_{Ds} and N_{Us} (as expressed in Eq. 7) from which N_n^p outputs the updated node feature descriptor h_v^k that is utilized in the $k+1$ iteration to produce updated edge descriptor. N_n^p is a four-layered MLP to which we append the fully connected layer in order to tune its weights for the nuclei identification task. Moreover, we generate the training samples for N_n^p by combining the output of N_{Us} and N_{Ds} on the training scans for each dataset. Afterward, we remove the fully connected layer from N_n^p, and use it as a feature extractor for generating updated node descriptors across each iteration k.

TABLE II
COMPUTING THE TP, FN, FP, AND TN FOR A NUCLEI CATEGORY X (I.E., TP_X, FN_X, FP_X, AND TN_X), FROM THE LIST OF PAIRED GROUND-TRUTHS

True Category	Predicted Category	Metric
X	X	TP_X
X	Not X	FN_X
Not X	X	FP_X
Not X	Not X	TN_X

V. EXPERIMENTAL SETUP

This section presents an overview of the datasets, the training and implementation protocols, as well as the evaluation metrics.

A. Datasets

We validated the proposed NGR scheme on four CRC histology datasets, dubbed CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7]. Due to space constraints, we refer the readers to each dataset paper for their detailed explanation.

B. Training and Implementation Protocol

The proposed scheme is implemented in Anaconda with Python 3.7.8, TensorFlow 2.3.0, and MATLAB R2020b on a machine with Intel Core i9-10940@3.3 GHz processor, 128 GB RAM, NVIDIA RTX 3090, cuDNN v7.5, and a CUDA Toolkit v11.0.221. The seven models (as described in Table I) are trained for 100 epochs across each dataset using categorical cross-entropy loss function, where 10% of the training scans were used for validation purposes. The optimizer used during the training was ADAM [25], which had an initial learning rate of 0.001. The source code of the proposed scheme will be released upon paper acceptance.

C. Evaluation Metrics

To evaluate the proposed scheme, we used the same evaluation metrics which are adopted by the other competitive approaches, i.e., the F-scores. To compute the F-score for each nuclei category, we first performed the pairings between the ground-truth nuclei and the detected nuclei using Kuhn–Munkres algorithm [26]. The pairing associates, to each ground-truth nuclei, the closest detected nuclei lying within a predefined radius. Moreover, it also returns the unpaired ground truths (ground-truth nuclei having no detected nuclei in the aforementioned predefined neighborhood) and the unpaired detected nuclei (unassigned detected nuclei). From these three lists, we examine the elements for each cell category and compute the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), as per the criteria defined in Table II. The unpaired ground truth and the unclassified detected nuclei are, respectively, considered as false negatives (in the sense that they should not have been missed) and false positives (in the sense that they should not have been detected). Afterward, we compute the F-score to evaluate the

TABLE III

DETERMINING THE OPTIMAL NODE ENCODER. FOR EACH NUCLEI TYPE ACROSS EACH DATASET, THE BEST PERFORMANCE IS IN BOLDFACE ACROSS EACH ROW, WHILE THE SECOND-BEST PERFORMANCE IS IN BLUE COLOR. THE ABBREVIATIONS ARE MET: METRIC, NP: NUMBER OF PARAMETERS, RST: REST EFFICIENT TRANSFORMER [27], R-50: RESNET-50 [28], MNV2: MOBILENETV2 [29], COT: CONTEXTUAL TRANSFORMER [30], MAT: MULTISCALE ADAPTIVE TRANSFORMER [31], TCC: FUSED TRANSFORMER AND CROSS CORRELATION MODEL [32], AND KTR: KINSHIP TRANSFORMER [33]

CRCHistoPhenotypes [4]							
Met	RST	R-50	MNv2	COT	MAT	TCC	KTR
F_{ep}	0.559	**0.572**	0.561	0.570	0.564	0.562	0.558
F_f	0.497	**0.527**	0.511	0.524	0.515	0.510	0.506
F_i	0.543	**0.571**	0.554	0.568	0.562	0.557	0.551
F_o	0.161	**0.222**	0.184	0.220	0.213	0.209	0.202
F_a	0.440	**0.473**	0.452	0.470	0.463	0.459	0.454
NP	37.4M	16.8M	**6.91M**	23.8M	18.4M	16.9M	43.5M
CoNSeP [5]							
F_{ep}	0.712	**0.739**	0.716	0.733	0.729	0.726	0.721
F_f	0.561	**0.588**	0.567	0.584	0.580	0.574	0.569
F_i	0.633	**0.656**	0.643	0.650	0.647	0.642	0.637
F_o	0.544	**0.564**	0.545	0.560	0.556	0.550	0.545
F_a	0.612	**0.636**	0.617	0.631	0.628	0.623	0.618
NP	38.6M	16.4M	**6.87M**	24.1M	18.7M	16.6M	43.9M
PanNuke [24]							
F_n	0.675	**0.686**	0.672	0.681	0.679	0.674	0.669
F_i	0.578	**0.589**	0.576	0.583	0.580	0.573	0.568
F_c	0.549	**0.563**	0.541	0.557	0.551	0.546	0.541
F_{de}	0.134	**0.155**	0.136	0.150	0.143	0.139	0.136
F_{ep}	0.571	**0.585**	0.572	0.581	0.574	0.567	0.562
F_a	0.626	**0.515**	0.499	0.510	0.505	0.499	0.495
NP	48.5M	22.6M	**9.43M**	33.4M	24.8M	22.9M	54.8M
Lizard [7]							
F_{ne}	0.560	**0.568**	0.557	0.565	0.563	0.559	0.552
F_{ep}	0.451	**0.463**	0.446	0.461	0.457	0.451	0.448
F_l	0.457	**0.477**	0.463	0.466	0.461	0.454	0.450
F_p	0.484	**0.493**	0.484	0.490	0.486	0.481	0.469
F_e	0.341	**0.352**	0.346	0.348	0.343	0.338	0.335
F_c	0.438	**0.450**	0.442	0.448	0.442	0.438	0.433
F_a	0.455	**0.467**	0.456	0.463	0.458	0.453	0.447
NP	53.7M	26.7M	**11.8M**	39.6M	28.9M	26.9M	57.5M

classification of each nucleus as per the standards laid down by the state-of-the-art [5]:

$$F_c = \frac{2(TP_{\mathrm{X}}+TN_{\mathrm{X}})}{2(TP_{\mathrm{X}}+TN_{\mathrm{X}})+\alpha_0 FP_{\mathrm{X}}+\alpha_1 FN_{\mathrm{X}}+\alpha_2 FP_{\mathrm{d}}+\alpha_3 FN_{\mathrm{d}}}, \tag{8}$$

where $\mathrm{FP_d}$ denotes the detection of false positives (unpaired detected nuclei), $\mathrm{FN_d}$ represents the detection of false negatives (unpaired ground truth nuclei), $\alpha_{0,1,2,3} = [2, 2, 1, 1]$ represents the metric hyperparameters selected as per the standards [5], $\mathrm{TP_X}$, $\mathrm{FN_X}$, $\mathrm{FP_X}$, and $\mathrm{TN_X}$ denotes the TP, FN, FP, and TN for nuclei category X, respectively. Moreover, we also computed F_{avg} that represents the average of the F-score computed for all the nuclei types. Apart from this, we also used the success rate in order to analyze the extent of homogeneity that the proposed scheme honors towards retaining the correctly identified nodes before the loose nodes assignment (see Section VI-D for more details).

VI. RESULTS AND DISCUSSION

This section presents a detailed evaluation of the proposed scheme and its comparison with state-of-the-art works. In addition, this section discusses an exhaustive set of ablation experiments through which we determined the hyperparameters of the proposed scheme. Lastly, this section presents a detailed discussion on the performance interpretation of the proposed approach.

A. Ablation Studies

We performed a series of ablation experiments to determine 1) The optimal node encoder, 2) The optimal stopping criterion for the NGR scheme, 3) The effect of reversing the traversal path direction, 4) The fusion of the results generated from the neural graph refinements at two levels (i.e., before and after loose nodes assignment), 5) The best detection model, 6) The effect of the initial node on the traversing path, 7) The compatibility of the proposed NGR featurization block with different graph partitioning schemes, and 8) Ability to integrate NGR featurization block with the popular clustering schemes.

1) The Optimal Node Encoder: The first ablation experiment is related to determining the model N, which gives the optimal performance towards extracting the node features for the accurate recognition of nuclei community within the WSI patches. For this purpose, we trained the feature extractor of different convolutional and transformer models, such as ResT [27], ResNet-50 [28], MobileNetv2 [29], Contextual Transformer (COT) [30], Multiscale Adaptive Transformer (MAT) [31], Fused Transformer and Cross Correlation Model (TCC) [32], and Kinship Transformer (KTR) [33]. Afterward, we integrated them within the proposed NGR scheme to generate the accurate latent representation of nuclei community. From Table III, we can observe that both COT [30] and ResNet-50 [28] can be selected as a node encoder within the proposed scheme to generate distinct feature representations. However, since ResNet-50 [28] offers a better trade-off between performance and computational complexity than COT [30], we chose ResNet-50 [28] as an optimal node encoder N in the rest of the experimentation.

2) The Optimal Stopping Criteria: In the proposed NGR scheme, we consider that the algorithm converges when the changes in the graph adjacency matrix remain below a certain threshold (σ) over a window of iterations of length K. We evaluate the change in the adjacency matrix ($\mathbb{A}^m$) as follows:

$$\sigma = \frac{\sum_{j=2}^{i_t} \mathbb{A}_j^m \oplus \mathbb{A}_{j-1}^m}{n_e}, \tag{9}$$

where n_e is the normalization term representing the number of edges within the initial graph $\mathbb{G}$. Table IV reports the overall performance and the convergence time obtained over different combinations threshold σ and window length K for each dataset. From Table IV, we can observe that increasing K increases the classification performance of the proposed scheme. However, it also drastically increases the computational time of the proposed approach. Similarly, by increasing the convergence threshold σ, we can get a significant boost in computational efficiency but at the expense of extreme deterioration in the classification performance. Therefore, through rigorous evaluation across each dataset, we selected $K = 5$ for $\sigma = 1\%$ to achieve convergence, as these values give optimal trade-off between classification performance and computational efficiency.

TABLE IV

Overall Performance and Convergence Iterations for Different Combinations of the σ and K Across Each Dataset. Performance Is Represented as an Average F-Score for All Nuclei Types. The Optimal Combination Is Shown in Boldface for Each Dataset. Moreover, the Abbreviations Are: RE: Recognition Performance, CG: Convergence (in Terms of iterations), CRCHisto: CRCHistoPhenotypes [4]

Type	Dataset	K/σ	1%	2%	3%	4%	5%
RE	CRCHisto	5	**0.473**	0.421	0.375	0.328	0.271
		10	0.479	0.429	0.382	0.331	0.280
		15	0.484	0.438	0.396	0.342	0.288
		20	0.507	0.446	0.403	0.354	0.296
		25	0.529	0.471	0.414	0.367	0.309
	CoNSeP	5	**0.636**	0.552	0.497	0.428	0.363
		10	0.640	0.569	0.526	0.457	0.398
		15	0.642	0.584	0.541	0.492	0.424
		20	0.644	0.593	0.563	0.511	0.456
		25	0.649	0.601	0.572	0.526	0.473
	PanNuke	5	**0.515**	0.487	0.472	0.463	0.456
		10	0.521	0.493	0.478	0.469	0.462
		15	0.526	0.502	0.483	0.476	0.468
		20	0.529	0.511	0.489	0.482	0.473
		25	0.532	0.523	0.493	0.489	0.481
	Lizard	5	**0.467**	0.441	0.428	0.416	0.403
		10	0.471	0.448	0.435	0.421	0.409
		15	0.475	0.454	0.441	0.428	0.413
		20	0.481	0.460	0.448	0.432	0.419
		25	0.484	0.465	0.454	0.439	0.427
CG	CRCHisto	5	**19**	13	10	7	4
		10	24	18	15	12	9
		15	29	23	20	17	14
		20	34	28	25	22	19
		25	39	33	30	27	24
	CoNSeP	5	**23**	19	13	8	6
		10	28	24	18	13	11
		15	33	29	23	18	16
		20	38	34	28	23	21
		25	43	39	33	28	26
	PanNuke	5	**37**	29	18	11	8
		10	42	34	23	16	13
		15	47	39	28	21	18
		20	52	44	33	26	23
		25	57	49	38	31	28
	Lizard	5	**44**	36	27	21	16
		10	49	41	32	26	21
		15	54	46	37	31	26
		20	59	51	42	36	31
		25	64	56	47	41	36

TABLE V

Determining the Performance of the Proposed Scheme When the Traversal Paths Are Reversed. For Each Dataset, the 'Actual' Column Represents the Mean Values of the Nuclei Types Obtained Using the Actual Traversal Path, While the 'Reversed' Column Shows the Mean Values of the Nuclei Obtained When the Traversal Path Was Reversed. The Last Column Shows the Disparity Score (d_s). Moreover, the Abbreviation of 'CRCHisto' Stands for 'CRCHistoPhenotypes [4]'. The Average Score for Each Dataset Is in Boldface

Dataset	Nuclei Type	Actual	Reversed	d_s
CRCHisto [4]	Epithelial	0.571	0.570	9.6x10^{-7}
	Fibroblast	0.530	0.523	7.2x10^{-6}
	Inflammatory	0.569	0.564	4.8x10^{-6}
	Others	0.223	0.217	1.4x10^{-5}
	Average	**0.473**	**0.468**	**6.7x10^{-6}**
CoNSeP [5]	Epithelial	0.736	0.732	3.1x10^{-6}
	Fibroblast	0.590	0.582	7.7x10^{-6}
	Inflammatory	0.661	0.656	4.3x10^{-6}
	Others	0.565	0.558	7.1x10^{-6}
	Average	**0.638**	**0.632**	**5.5x10^{-6}**
PanNuke [24]	Neoplastic	0.684	0.678	7.7x10^{-7}
	Inflammatory	0.591	0.583	1.2x10^{-6}
	Connective	0.565	0.559	9.3x10^{-7}
	Dead	0.152	0.141	6.3x10^{-6}
	Epithelial	0.584	0.578	9.1x10^{-7}
	Average	**0.515**	**0.507**	**2.0x10^{-6}**
Lizard [7]	Neutrophil	0.564	0.561	6.1x10^{-8}
	Epithelial	0.469	0.462	1.7x10^{-7}
	Lymphocyte	0.481	0.473	1.9x10^{-7}
	Plasma	0.494	0.489	1.1x10^{-7}
	Eosinophil	0.348	0.342	2.0x10^{-7}
	Connective	0.453	0.446	1.7x10^{-7}
	Average	**0.468**	**0.462**	**1.5x10^{-7}**

3) Reversing the Traversal Path: We saw in section IV-B.3 that the edge descriptors, when traversing the path, are not commutative. We wanted to investigate here to what extent this property affects the segmentation when the traversal directions are reversed. For this purpose, we compared the performances obtained with the paths extracted in the previous section with its counterpart in the reversed direction. Moreover, we evaluated the disparity between both variants as follows:

$$d_s = \frac{1}{n_{paths}} \sum_{i=1}^{n_{paths}} \frac{|F_{\mathbb{P}} - F_{\mathbb{P}_r}|}{F_{\mathbb{P}}}, \tag{10}$$

where d_s denotes the disparity measure, n_{paths} denotes the number of paths, $F_{\mathbb{P}}$ denotes the average of F-score across the nuclei type while traversing the actual path $\mathbb{P}$, and $F_{\mathbb{P}_r}$ denotes the average of F-score for each nuclei type when the path $\mathbb{P}$ is reversed. From Table V, we can observe that across all the datasets, reversing the path $\mathbb{P}$ does not affect much the overall performance of the proposed scheme as reflected in the low values of d_s for each dataset. Therefore, we can conclude that alternating the path does not affect the proposed scheme's nuclei recognition performance.

4) Fusion of Graph Refinements: Given the fact that graph segmentation can be performed from any traversal path, we wanted to explore potential enhancement by fusing the outcomes of the classifications issued from different traversal paths. This procedure can be viewed as an ensemble classification approach. We performed this ablation as follows: We selected M number of paths (where M is taken as an odd number) starting at different locations using the same procedure in Section IV-B.2, then we fused the classification outcomes by maximum voting at two levels: a) before loose nodes re-assignment, and b) after loose nodes re-assignment. Figure 4 reports the obtained results across the CRCHistoPhenotypes [4], and CoNSeP [5] datasets. We can observe here that the fusion between the number of paths (n_{paths}) led to the increase in nuclei recognition across both datasets, and without reassigning loose nodes, we also achieve a decent performance by fusing the outputs from 15 traversal paths (i.e., when $n_{paths} = 15$). However, such an operation also requires a drastic amount of time that can be avoided by reassigning the loose nodes based on latent space associations (see Section IV-C.3). Similarly, from Table IX and Figure 4-1st Column, we can observe that with reassigning the loose nodes, the performance of $n_{paths} = 15$ during graph fusion is close to its conventional traversal with $n_{paths} = 1$. As $n_{paths} = 1$ scheme takes a significantly lesser amount of convergence time than $n_{paths} = 15$, in the rest of the experimentation, we proceeded with recognizing the nuclei community with $n_{paths} = 1$.

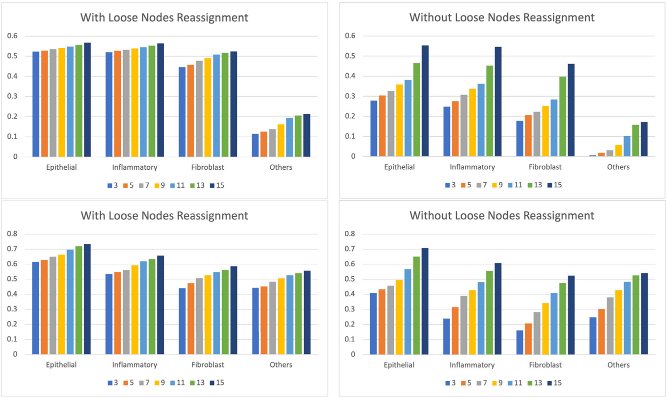

Fig. 4. Performance evaluation of the proposed scheme in terms of F-score when the outcomes from different traversal paths are fused. The first column represents the performance without loose node reassignment, and the second column shows the performance with loose node reassignment. Moreover, the first row shows the results on the CRCHistoPhenotypes [4], and the second row shows the results on the CoNSeP [5] dataset. Also, in each plot, the color for the bars depicts the different number of fused paths (e.g., 3, 5, 7, 9, 11, 13, and 15).

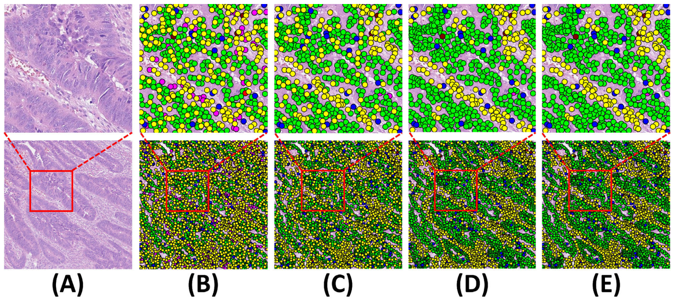

Fig. 5. Performance evaluation of the proposed framework to accurately recognize densely distributed nuclei communities within the WSI patch as compared to state-of-the-art methods. The first column shows the example patch. Columns 2, 3, and 4 show the results of Hover-Net [5], NCD [10], and the proposed framework, respectively. The fifth column shows the ground truth labels.

5) The Best Detection Model: This ablative experiment is related to determining the optimal pre-trained model that gives the best initial nuclei detection and classification performance across each dataset. For this purpose, we selected the models such as Hover-Net [5], Mask R-CNN [23], and Faster R-CNN [22], as they have also been used in the literature for the nucleus detection tasks [5], [34]. It should also be noted that the results for this experiment are computed using the centroid location of each nucleus region, as well as their predicted labels. From Table VI, we can observe that the Hover-Net [5] gives the optimal detection performance as compared to detection models in terms of F-scores. The performance improvements achieved by the Hover-Net [5] are because of its novel multi-scale contextual and residual learning blocks [5] that enable the accurate localization of the nuclei regions even in similarly textured WSI patches. Therefore, in the rest of the experimentation, we used Hover-Net [5] as a detection backbone within the proposed scheme.

TABLE VI

Determining the Detection Model That Gives Best Localization of the Nuclei Regions in Terms of F-Scores. Bold Indicates the Best Scores, While the Second-Best Scores Are in Blue Color. Moreover, the Abbreviations Are: HNet: Hover-Net [5], M-RN: Mask R-CNN [23], F-RN: Faster R-CNN [22]

Dataset	Nuclei Type	HNet	M-RN	F-RN
CRCHisto [4]	Epithelial	0.486	**0.503**	0.499
	Inflammatory	**0.573**	0.537	0.534
	Fibroblast	**0.302**	0.294	0.298
	Others	**0.178**	0.077	0.081
	Average	**0.384**	0.352	0.353
CoNSeP [5]	Epithelial	**0.635**	0.595	0.592
	Inflammatory	**0.631**	0.590	0.586
	Fibroblast	**0.566**	0.520	0.527
	Others	**0.426**	0.098	0.105
	Average	**0.564**	0.450	0.452
PanNuke [24]	Epithelial	0.513	**0.520**	0.517
	Inflammatory	**0.525**	0.500	0.498
	Neoplastic	**0.648**	0.590	0.586
	Connective	**0.559**	0.420	0.417
	Dead	0.142	0.220	**0.224**
	Average	**0.477**	0.450	0.448
Lizard [7]	Epithelial	0.354	**0.433**	0.429
	Neutrophil	**0.548**	0.250	0.245
	Connective	**0.436**	0.396	0.389
	Lymphocyte	**0.441**	0.413	0.411
	Plasma	**0.339**	0.289	0.286
	Eosinophil	**0.248**	0.075	0.079
	Average	**0.394**	0.309	0.306

6) Effect of the Initial Node Selection: In this study, we wanted to investigate the effect of the initial node location of the traversal path on the overall nuclei recognition performance. For this purpose, we computed the mean and the standard deviations of the performance metrics obtained over a large number of paths starting at different locations of the community graph. To ensure the set of initial nodes spans the whole graph while avoiding redundancy, we subsampled the nodes, at the rate of five samples, along a firstly extracted path $\mathbb{P}$, and measured the classification performance in terms of F-score for each sub-path generated from each of the sampled nodes. Afterward, we took the mean and standard deviation of the scores obtained across all the sub-paths, where the total number of sub-paths is equal to the number of sampled nodes. The number of sampled nodes is determined by dividing the total nodes in $\mathbb{P}$ by the sampling factor (which is 5). Moreover, the performance evaluation in terms of mean and standard deviation across each dataset is shown in Table VII. We can observe here that, across all four datasets, the recognition performance of the proposed scheme in terms of F-score remains consistent, and there is a very much degree of variability which was observed towards recognizing different nuclei communities across all the datasets. For example, on CRCHistoPhenotypes [4], the proposed scheme achieved 1.2% variability for the average F-score in terms of the ratio of standard deviation over a mean (RoSM). Similarly, across CoNSeP [5], PanNuke [24], and Lizard [7] datasets, the proposed framework achieved the 0.7%, 0.9%, and 1.0% variability, respectively, in terms of the RoSM metric for the average F-score. These low values of RoSM indicate the stability of the proposed approach and highlight the fact that the proposed scheme is virtually independent of the choice of the initial node for the graph traversing.

7) NGR Featurization With Graph Partitioning Methods: In this series of experimentation, we tested one of the main features of the proposed NGR scheme, i.e., its capacity to integrate with the popular graph partitioning methods. The working principle of the NGR scheme can be broken down into two phases, where in the first phase, the node and edge features, generated through the trained MLPs, are propagated

TABLE VII

DETERMINING THE EFFECT ON THE CHOICE OF THE INITIAL NODE IN TERMS OF F-SCORES. THE AVERAGE RECOGNITION SCORE ACROSS EACH DATASET IS IN BOLDFACE. MOREOVER, 'STD' STANDS FOR 'STANDARD DEVIATION'

Dataset	Nuclei Type	HNet	M-RN	F-RN
CRCHisto [4]	Epithelial	0.571	0.009	0.015
	Fibroblast	0.530	0.006	0.011
	Inflammatory	0.569	0.005	0.008
	Others	0.223	0.007	0.031
	Average	**0.473**	**0.006**	**0.012**
CoNSeP [5]	Epithelial	0.736	0.006	0.008
	Fibroblast	0.590	0.002	0.003
	Inflammatory	0.661	0.009	0.013
	Others	0.565	0.006	0.010
	Average	**0.638**	**0.005**	**0.007**
PanNuke [24]	Neoplastic	0.684	0.003	0.004
	Inflammatory	0.591	0.007	0.011
	Connective	0.565	0.006	0.010
	Dead	0.152	0.008	0.052
	Epithelial	0.584	0.005	0.008
	Average	**0.515**	**0.005**	**0.009**
Lizard [7]	Neutrophil	0.564	0.004	0.007
	Epithelial	0.469	0.007	0.014
	Lymphocyte	0.481	0.002	0.004
	Plasma	0.494	0.004	0.008
	Eosinophil	0.348	0.006	0.017
	Connective	0.453	0.009	0.019
	Average	**0.468**	**0.005**	**0.010**

within the whole graph $\mathbb{G}$. The combination of these features is then used in the second phase by the edge classifier to predict whether the respective edge, connecting two adjacent nodes, should be pruned, or retained. The first phase typically conducts the featurization process, whereas the second phase is responsible for retaining the purest (homogeneous) clusters of nuclei communities. Moreover, the loose nodes are then reclassified to their accurate cluster by analyzing their latent space homogeneity with each of the nuclei groups using Euclidean distance.

Instead of using the edge classifier, we can also integrate NGR featurization block with the traditional graph partitioning schemes, such as normalized-cuts (N-Cuts) [35]. For this purpose, we first define the association between the adjacent nodes v and u in graph $\mathbb{G}$ as:

$$A_s^{(v,u)} = \sqrt{\sum_{i=0}^{n_f-1} \left(h_v^k(i) - h_u^k(i)\right)^2}, \tag{11}$$

where n_f denotes the number of features, generated by NGR featurization block, for node v and u. Afterward, we define a cut N_c with the cost $n_{cost}^{v,u}$, as expressed below:

$$n_{cost}^{(v,u)} = \|h_{cost}^{(v,u)} + e_{cost}^{(v,u)}\|, \tag{12}$$

$h_{cost}^{(v,u)}$ represents the cost produced by the nodes v, and u, and $e_{cost}^{(v,u)}$ is the cost of the respective edge that joins nodes v, and u. $\|.\|$ denotes the linear normalization operator. $h_{cost}^{(v,u)}$ and $e_{cost}^{(v,u)}$ are computed as:

$$h_{cost}^{(v,u)} = \frac{A_s^{(v,u)}}{\sum_{m,n\in V|m\neq n} A_s^{(m,n)}}, \tag{13}$$

$$e_{cost}^{(v,u)} = \frac{\sum_{i=0}^{e_f-1} e_{(v,u)}^k(i)}{\sum_{m,n\in V|m\neq n} \sum_{j=0}^{e_f-1} e_{m,n}^k(j)}, \tag{14}$$

where e_f denotes the number of edge features between adjacent nodes v and u. If $n_{cost}^{(v,u)} < t$, for each node pair v and u, then the link between them is retained. Otherwise, N_c cuts the link between v and u, that eventually partitions $\mathbb{G}$ into multiple sub-graphs. Also, t is a cut threshold that is empirically determined to be 0.5. Moreover, the loose nodes, produced by N_c, are assigned to their nearest nuclei group by analyzing their latent space homogeneity via Euclidean distance, as discussed in Section IV-C.3.

The results for this experiment are reported in Table VIII in which we can observe that coupling NGR with N-Cuts produces decent nuclei classification performance across all four datasets. For example, on CRCHistoPhenotypes [4] dataset, it produces an average F-score (F_a) of 0.427. Similarly, across CoNSeP [5], PanNuke [24], and Lizard [7] datasets, it produces an F_a score of 0.588, 0.477, and 0.435, respectively. Although, the performance of 'NGR + N-Cuts' approach is a bit lower than the original NGR scheme, which uses an edge classifier (EC), i.e., 'NGR + EC'. Nevertheless, the performance of 'NGR + N-Cuts' is still appreciable, which can be further improved by designing a better metric to represent node and edge associations. Additionally, noticing the fact that NGR featurization can be coupled with graph partitioning algorithm (such as N-Cuts [35]), we can explore the applicability of the proposed NGR scheme in a wide variety of applications that uses graph partitioning methods.

8) NGR Featurization With Clustering Methods: In these series of experiments, we explored the applicability of the NGR featurization block to be coupled with traditional clustering methods, such as K-means clustering. To integrate NGR featurization block with K-means clustering, we compute the node and edge features, and propagate them within $\mathbb{G}$ until convergence is achieved (as discussed in Algorithm 2). Afterward, for each node v connected to another node u, we generate their combined features $f_{v,u}$, such as $f_{v,u}$=[h_v^k, $e_{v,u}^k$, h_u^k], where h_v^k represents the node v features, $e_{v,u}^k$ denotes the features of edge adjacent to node v and u, and h_u^k represents node u features. The combined features are collected for each node pairs within $\mathbb{G}$, and are passed to the K-means clustering scheme. Here, we specify the number of clusters to be equal to the number of nucleus classes as defined in each dataset. Once K-means clustering distributes all the nodes to the specified number of clusters, we then assign the nuclei category to each cluster. The assignment of nuclei label to each cluster is performed in the following steps: 1) First we pick a single node n from each nuclei class for the given dataset. 2) We compute the initial node features h_n^0 of node n using the model N. 3) We find the Euclidean distance of node n with all the nodes in each cluster using the initial node features. Here, the initial node features (h_m^0) of node m in each cluster is also computed using N model. 4) If the node n has the minimum distance with node m, then the label of node n is assigned to the whole cluster that contains node m.

The results for coupling NGR featurization with K-means clustering is reported in Table VIII, where we can observe that when we feed the features computed through NGR to K-means clustering, we get a competitive nuclei classification performance across each dataset. For example, across CRCHistoPhenotypes [4] dataset, the 'NGR + K-means' produces an average F-score of 0.454 towards recognizing different nuclei types. Similarly, across CoNSeP [5], PanNuke [24], and Lizard [7] datasets, the 'NGR + K-means' achieved an average F-score of 0.615, 0.501, and 0.449, respectively.

TABLE VIII

Determining the Compatibility of the Featurization Block Within Proposed NGR Scheme With the Popular Graph Partitioning and Clustering methods. The Performance Evaluation Is Performed in Terms of F-score Across Each dataset. Bold Indicates the Best scores, While the second-best Scores Are underlined. Moreover, the Abbreviations are: 'NGR + EC': the Originally Proposed NGR Scheme Which Uses an Edge classifier, 'NGR + N-Cuts': the Proposed NGR Featurization Block That Is Coupled With the Traditional Normalized Cut Partitioning scheme, 'NGR + K-means': the Proposed NGR Featurization Block That Is Integrated With the Traditional K-means Clustering Scheme for Nuclei classification

Nuclei Type	NGR + EC	NGR + N-Cuts	NGR + K-means
CRCHistoPhenotypes [4]			
Epithelial	**0.572**	0.523	0.564
Inflammatory	**0.571**	0.514	0.552
Fibroblast	**0.527**	0.492	0.508
Others	**0.222**	0.179	0.193
Average	**0.473**	0.427	0.454
CoNSeP [5]			
Epithelial	**0.739**	0.685	0.714
Inflammatory	**0.659**	0.626	0.642
Fibroblast	**0.589**	0.524	0.568
Others	**0.564**	0.518	0.539
Average	**0.637**	0.588	0.615
PanNuke [24]			
Epithelial	**0.585**	0.541	0.575
Inflammatory	**0.589**	0.558	0.572
Neoplastic	**0.686**	0.643	0.667
Connective	**0.563**	0.526	0.551
Dead	**0.155**	0.121	0.142
Average	**0.515**	0.477	0.501
Lizard [7]			
Epithelial	**0.463**	0.429	0.452
Neutrophil	**0.568**	0.537	0.541
Connective	**0.450**	0.428	0.437
Lymphocyte	**0.477**	0.433	0.460
Plasma	**0.493**	0.471	0.483
Eosinophil	**0.352**	0.314	0.324
Average	**0.467**	0.435	0.449

Although, the performance of 'NGR + K-means' approach is lagging behind the original 'NGR + EC' scheme by 4.01%, 3.45%, 2.71%, and 3.85% in terms of average F-score across CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7] datasets, respectively. But it is still appreciable because 'NGR + K-means' also outperforms state-of-the-art methods, such as Hover-Net [5] and NCD [10], along with proposed 'NGR + EC' approach, as evident from Tables VIII and IX. Furthermore, observing the fact that NGR scheme can be coupled with the traditional clustering methods opens up a new horizon for NGR to be applied to the wide variety of clustering applications.

However, keeping in mind the scope of this research, we proceed with using the 'NGR + EC' scheme in the rest of experimentation as it gives better performance than 'NGR + N-Cuts' and 'NGR + K-means' variants toward performing the nuclei classification tasks.

B. Quantitative Comparison

In this section, we report a thorough quantitative comparison of the proposed scheme with state-of-the-art works across the four datasets. From Table IX, we can observe that the Hover-Net [5] model gives a relatively better nuclei recognition as compared to other models, and when we couple Hover-Net [5] with the proposed NGR scheme and use it as an initial detection and classification backbone, we achieved significant performance improvements as compared to the state-of-the-art in terms of F-score. For example, on CRCHistoPhenotypes [4], we achieved 10.31%, 14.23%, and 19.81% improvements over the state-of-the-art in terms of F-score for recognizing epithelial, fibroblast, and miscellaneous (others) nuclei, respectively. Similarly, on the CoNSeP dataset [5], we achieved 9.47%, 3.64%, 3.39%, and 16.48% improvements over the second-best method in terms of F-score for recognizing epithelial, inflammatory, fibroblast, and miscellaneous (others) nuclei, respectively. On the PanNuke dataset [24], the proposed scheme achieved 10.59%, 8.14%, 4.81%, and 1.59% over state-of-the-art for extracting epithelial, inflammatory, neoplastic, and connective nuclei, respectively in terms of F-score. Also, on Lizard dataset [7], the proposed scheme achieved 6.47%, 2.99%, 1.77%, 8.38%, 20.68%, and 24.14% over state-of-the-art for extracting epithelial, neutrophil, connective, lymphocyte, plasma, and eosinophil nuclei, respectively in terms of F-score. Similarly, in another series of experiments, we compared the classification performance of the proposed NGR scheme with state-of-the-art graph-based nucleus classification models, such as HAT-Net [36], and Structure Embedded Nucleus Classification (SENC) model [37]. The performance comparison is reported in Table X from which we can observe that across all four datasets, the proposed NGR scheme produces the best classification performance as compared to its competitors. For example, on CRCHistoPhenotypes [4], the proposed method achieved 10.99% improvements in terms of an average F-score (F_a). Similarly, across CoNSeP [5], PanNuke [24], and Lizard [7] datasets, the proposed NGR scheme achieved 6.59%, 5.82%, and 2.99% performance improvements over state-of-the-art methods in terms of F_a. These performance improvements stem from the fact that the proposed scheme exploits the latent homogeneity between the shared nuclei features and refines them iteratively across the community graph nodes and edges. Such neural broadcasting allows the edge classifier to recognize the heterogeneity and homogeneity between the adjacent nodes to retain or prune the connections, respectively.

C. Qualitative Comparison

Figures 5, and 6 report the qualitative assessment of the proposed scheme towards recognizing the nuclei types as compared to state-of-the-art works. In Figure 5, we showcased an exemplar WSI patch from the Lizard dataset [7] that contains a large population of nuclei regions. Here, in the actual patch (shown in the second row), we cannot visually see the difference in the recognition. However, within the zoomed portion (as shown in the first row), we can see that the proposed scheme has significantly improved nuclei recognition rate as compared to the state-of-the-art Hover-Net [5], and NCD [10] approaches, considering the ground truth annotations. Similarly, in Figure 6, we report some qualitative examples from all four datasets that showcase the recognition capacity of the proposed framework in light of the ground truth annotations. Figure 6 also demonstrate the fact that the proposed scheme has remarkably outperformed the state-of-the-art Hover-Net [5] and NCD [10] towards

TABLE IX

Evaluation of the Proposed Scheme With State-of-the-Art in Terms of F-Score. For Each Nuclei Type, the Best and Second-Best Scores Are in Boldface and Blue Color Within Each Row, Respectively. '-' Indicates That the Given Method Has Not Been Validated on the Respective Dataset. The Abbreviations Are: EMS: Entropy Maximized Scale [3], SLD: Sparse Learned Dictionaries [1], NCD: Nucleus Community Detection [10], MCC: Multiplex Cellular Communities [2], HNet: Hover-Net [5], M-RN: Mask R-CNN [23], and PS: Proposed Scheme (Backboned With Hover-Net [5])

Metric	EMS	SLD	NCD	MCC	HNet	M-RN	PS
CRCHistoPhenotypes [4]							
F_{ep}	0.218	0.234	0.513	0.346	0.486	0.503	**0.572**
F_{i}	0.154	0.206	0.549	0.385	**0.573**	0.537	0.571
F_{f}	0.102	0.137	0.452	0.193	0.302	0.294	**0.527**
F_{o}	0.056	0.081	0.162	0.107	0.178	0.077	**0.222**
F_{a}	0.132	0.164	0.419	0.257	0.384	0.352	**0.473**
CoNSeP [5]							
F_{ep}	0.352	0.414	0.669	0.421	0.635	0.595	**0.739**
F_{i}	0.298	0.336	0.635	0.338	0.631	0.590	**0.659**
F_{f}	0.206	0.249	0.569	0.276	0.566	0.520	**0.589**
F_{o}	0.135	0.181	0.471	0.212	0.426	0.098	**0.564**
F_{a}	0.247	0.295	0.586	0.311	0.564	0.450	**0.637**
PanNuke [24]							
F_{ep}	0.236	0.281	0.523	0.346	0.513	0.520	**0.585**
F_{i}	0.147	0.154	0.541	0.273	0.525	0.500	**0.589**
F_{n}	0.112	0.130	0.653	0.219	0.648	0.590	**0.686**
F_{c}	0.073	0.106	0.554	0.168	0.559	0.420	**0.563**
F_{de}	0.094	0.113	0.147	0.171	0.142	**0.220**	0.155
F_{a}	0.132	0.156	0.483	0.235	0.477	0.450	**0.515**
Lizard [7]							
F_{ep}	0.198	0.252	0.428	0.318	0.354	0.433	**0.463**
F_{ne}	0.127	0.186	0.551	0.249	0.548	0.250	**0.568**
F_{c}	0.086	0.134	0.442	0.205	0.436	0.396	**0.450**
F_{l}	0.104	0.167	0.437	0.227	0.441	0.413	**0.477**
F_{p}	0.059	0.098	0.391	0.162	0.339	0.289	**0.493**
F_{e}	0.004	0.041	0.267	0.103	0.248	0.075	**0.352**
F_{a}	0.096	0.146	0.419	0.210	0.394	0.309	**0.467**

TABLE X

Performance Evaluation of the Proposed NGR Scheme With state-of-the-art Graph-Based Nuclei Classification methods. Bold Indicates the Best scores, While the second-best Performance Is in Blue color. Moreover, the Abbreviations Are: HAT-Net: Hierarchical Transformer Graph Neural Network [36], SENC: Structure Embedded Nucleus Classification Model [37]

Nuclei Type	NGR (Proposed)	HAT-Net [36]	SENC [37]
CRCHistoPhenotypes [4]			
Epithelial	**0.572**	0.491	0.523
Inflammatory	**0.571**	0.554	0.548
Fibroblast	**0.527**	0.327	0.419
Others	**0.222**	0.196	0.196
Average	**0.473**	0.392	0.421
CoNSeP [5]			
Epithelial	**0.739**	0.639	0.646
Inflammatory	**0.659**	0.628	0.632
Fibroblast	**0.589**	0.570	0.592
Others	**0.564**	0.473	0.510
Average	**0.637**	0.577	0.595
PanNuke [24]			
Epithelial	**0.585**	0.538	0.564
Inflammatory	**0.589**	0.546	0.552
Neoplastic	**0.686**	0.661	0.648
Connective	**0.563**	0.572	0.516
Dead	**0.155**	0.144	0.151
Average	**0.515**	0.492	0.485
Lizard [7]			
Epithelial	**0.463**	0.415	0.454
Neutrophil	**0.568**	0.549	0.549
Connective	**0.450**	0.443	0.446
Lymphocyte	**0.477**	0.456	0.462
Plasma	**0.493**	0.438	0.471
Eosinophil	**0.352**	0.322	0.339
Average	**0.467**	0.437	0.453

accurately recognizing the nuclei communities across all four datasets. This margin of improvements over state-of-the-art stems from the ability of the proposed scheme to distinguish the contextual, semantic, and spatial features of the nuclei regions contained within the histology patches, which results in the separation of the heterogeneous and homogeneous nodes during NGR iterations. Afterward, assigning the loose nuclei to their accurate distribution by finding the minimum distance in the latent domain allows the proposed scheme to drastically improve the nuclei communities recognition rates, especially in the dense nuclei community regions, as evident from the examples shown in Figures 5, and 6.

D. Performance Interpretation

The increased performance of the proposed NGR scheme stems from its strong capacity for deriving homogeneous nuclei communities. The NGR scheme, through its tenacious edge pruning, massively reduces the number of nodes (i.e., nuclei) in each community, keeping the most "pure" group that exhibits less number of outliers (misclassified nodes). This phenomenon can be evidenced in Table XI where columns 2 to 4 show the initial detection performance, in terms of success rate, before applying the NGR refinements, and columns 5 to 7 report the performance of the proposed scheme on the retained (or filtered) nodes after applying the NGR refinements, and the last column, 8, shows the number of loose nodes after NGR. We can also observe from Table XI that the proposed NGR refinements only retain the purest (the most homogeneous nodes), and deletes about 65.06%, 65.34%, 61.39%, and 79.70% nodes on average across CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7] dataset, respectively. The deleted (or loose nodes) are mostly the ones that were misclassified during the initial detection phase, and by excluding them from the filtered nodes, we can notice the significant performance boost of 14.01%, 16.67%, 16.57%, and 18.34% on average, in terms of success rate, achieved by the proposed NGR refinements across CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7] dataset, respectively (see the performance difference in SRR and SRI columns within Table XI for each dataset). Moreover, in the subsequent stage, the high degree of homogeneity obtained with the NGR scheme acts positively on the loose node assignment by boosting/minimizing the likelihood of correct/false assignments.

In a nutshell, the proposed approach can be dubbed as a filter and aggregate method by analogy to the standard split and merge segmentation scheme. Indeed, in the first phase, the NGR scheme, through its iterations, keeps filtering out the outliers nodes till no further operation is possible (according to the stopping criterion). In the second phase, the loose nodes assignment performs a regrouping, using a simple vicinity metric, yet anchored by the homogeneous communities.

E. Computational Time and Complexity

Although the proposed scheme is quite robust in accurately recognizing the nuclei communities. But, due to the apparent

TABLE XI

Evaluation of the Homogeneity of the Nuclei Communities at the Initial Segmentation (Obtained Here by Applying Hover-Net, See Columns 2 to 4) and After Applying the Proposed NGR Scheme (Before the Loose Nodes Assignment, Columns 5 to 7) Across the Four Datasets. The Abbreviations Are: NT: Nuclei Types, TN: Total Nodes, CRNB: Correctly Recognized Nodes Before NGR Iterations, SRI: Success Rate For Initial Recognition, RN: Retained Nodes (After NGR Iterations), CRNA: Correctly Identified Retained Nodes (Obtained After NGR Iterations), SRR: Success Rate For Correctly Recognizing the Retained Nodes (Obtained After NGR Iterations), LN: Loose Nodes (Obtained After NGR Iterations), Ep: Epithelial, Fi: Fibroblast, In: Inflammatory, Ot: Others, Ne: Neoplastic, Co: Connective, De: Dead, Nu: Neutrophil, Ly: Lymphocyte, Pl: Plasma, Eo: Eosinophil, and μ: Average

CRCHistoPhenotypes [4]

NT	TN	CRNB	SRI	RN	CRNA	SRR	LN
Ep	1774	1351	0.7615	621	586	0.9436	1153
Fi	3088	2513	0.8137	865	798	0.9225	2223
In	3749	3034	0.8092	1449	1326	0.8845	2300
Ot	482	227	0.4709	192	185	0.9635	290
μ	9093	7125	0.7835	3177	2895	0.9112	5916

CoNSeP [5]

NT	TN	CRNB	SRI	RN	CRNA	SRR	LN
Ep	561	414	0.7379	197	181	0.9187	364
Fi	1638	1305	0.7967	803	743	0.9252	835
In	3214	2492	0.7753	965	868	0.8994	2249
Ot	3364	2346	0.6973	1077	935	0.8681	2287
μ	8777	6557	0.7470	3042	2727	0.8964	5735

PanNuke [24]

NT	TN	CRNB	SRI	RN	CRNA	SRR	LN
Ne	23.0k	17.8k	0.7724	11.0k	10.2k	0.9261	11.9k
In	9561	6927	0.7245	3538	3119	0.8815	6023
Co	16.0k	12.1k	0.7584	4648	4186	0.9006	11.3k
De	960	593	0.6177	231	194	0.8398	729
Ep	7236	5734	0.7924	2461	2253	0.9154	4775
μ	56.8k	43.2k	0.7603	21.9k	19.9k	0.9114	34.8k

Lizard [7]

NT	TN	CRNB	SRI	RN	CRNA	SRR	LN
Nu	4116	2824	0.6861	1194	1018	0.8525	2922
Ep	210k	148k	0.7058	46.2k	40.2k	0.8698	164k
Ly	92.2k	59.5k	0.6457	17.5k	13.0k	0.7443	74.7k
Pl	24.8k	19.6k	0.7921	5967	5381	0.9017	18.8k
Eo	2979	2137	0.7173	1133	984	0.8684	1846
Co	97.3k	60.7k	0.6236	15.5k	12.2k	0.7868	81.7k
μ	432k	293k	0.6793	87.6k	72.9k	0.8319	344k

Exact value representations are: 23.0k: 23,045, 16.0k: 16,027, 56.8k: 56,829, 17.8k: 17,802, 12.1k: 12,156, 43.2k: 43,212, 11.0k: 11,062, 21.9k: 21,940, 10.2k: 10,245, 19.9k: 19,997, 11.9k: 11,983, 11.3k: 11,379, 34.8k: 34,889, 210k: 210,372, 92.2k: 92,238, 24.8k: 24,861, 97.3k: 97,347, 432k: 431,913, 148k: 148,491, 59.5k: 59,562, 19.6k: 19,694, 60.7k: 60,715, 293k: 293,423, 46.2k: 46,282, 17.5k: 17,526, 15.5k: 15,573, 87.6k: 87,675, 40.2k: 40,259, 13.0k: 13,046, 12.2k: 12,253, 72.9k: 72,941, 164k: 164,090, 74.7k: 74,712, 18.8k: 18,894, 81.7k: 81,774, 344k: 344,238.

TABLE XII

Evaluation of the Proposed Scheme With state-of-the-art Methods in Terms of Computational time, complexity, and Detection performance. Bold Denotes the Best performance, While the second-best Is in Blue color. The Abbreviations are: F_A: Average F-score, R-Time: Running Time, F-RN: Faster R-CNN [22], Proposed: Proposed Scheme (Backboned With Hover-Net [5]), NCD: Nucleus Community Detection [10], HNet: Hover-Net [5], M-RN: Mask R-CNN [23], MCC: Multiplex Cellular Communities [2]

Metric	F-RN	Proposed	NCD	HNet	M-RN	MCC
CRCHistoPhenotypes [4]						
F_a	0.353	**0.473**	0.419	0.384	0.352	0.257
R-Time (s)	95.18	20.34	33.05	**10.92**	99.17	26.58s
GFLOPs	9.21	3.51	5.24	**3.04**	11.92	7.63
CoNSeP [5]						
F_a	0.452	**0.637**	0.586	0.564	0.450	0.311
R-Time (s)	97.26	22.83	36.52	**11.04**	106.98	28.01
GFLOPs	8.64	3.47	5.09	**2.89**	10.63	6.76
PanNuke [24]						
F_a	0.448	**0.515**	0.483	0.477	0.450	0.235
R-Time (s)	102.56	28.06	43.85	**19.63**	112.24	34.53
GFLOPs	10.26	4.89	6.74	**4.63**	13.67	8.94
Lizard [7]						
F_a	0.306	**0.467**	0.419	0.394	0.309	0.210
R-Time (s)	115.92	35.27	50.86	**24.52**	128.46	39.52
GFLOPs	13.45	7.64	9.56	**5.97**	18.32	11.37

computational complexity of the NGR pipeline, achieving full convergence across each dataset takes a bit of time on a machine described in Section V-B. For example, from Table XII, we can observe that the proposed NGR scheme takes around 20.34 seconds, 22.83 seconds, 28.06 seconds, and 35.27 seconds on average to process one scan from the CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7] datasets, respectively. By analyzing the run-time performance of the proposed NGR scheme, we can notice that it lags behind Hover-Net [5] across each dataset. However, this lag is expected because the proposed NGR scheme uses Hover-Net [5] as a backbone, so it's time complexity will be accumulated within the time performance of the proposed NGR scheme. Similarly, the computational complexity of the proposed scheme is expected to be greater than Hover-Net [5] in terms of GFLOPs (see Table XII). However, considering the trade-off between classification performance and computational complexity, the proposed NGR scheme significantly outperforms Hover-Net [5], as it provides 18.81%, 11.45%, 7.37%, and 15.63% better performance in terms of F_a across CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7] datasets, respectively. In addition to this, it also outperforms other state-of-the-art methods, such as MCC [2], and NCD [10] in terms of classification performance, computational complexity, and running time which is appreciable (see Table XII). Nevertheless, computational time is one of the limitations of the proposed scheme, and the remedies to improve it are discussed in Section VI-F.

F. Limitations

The two limitations of the proposed scheme are related to its computational time and its dependency on the existing detection models to procure the centroid of the nuclei regions. The first limitation, i.e., the computational time, can be addressed by further fine-tuning the system's hyper-parameters (e.g., the number of NGR iterations, the convergence window size, the convergence threshold) without compromising the performance, so to set the optimal trade-off between computational efficiency and the detection performance as per the clinical needs. Apart from this, dedicated hardware accelerators for graph processing can also be envisaged further to improve the execution time of the proposed scheme [38].

The second limitation of the proposed scheme is its dependency on computing the nuclei centroid within the WSI patches using the detection backbone. Here, if the detection backbone misses the localization of the nuclei region, then their labels can also not be refined by the proposed scheme. Therefore, we prefer integrating the proposed scheme with only the best-performing detection models

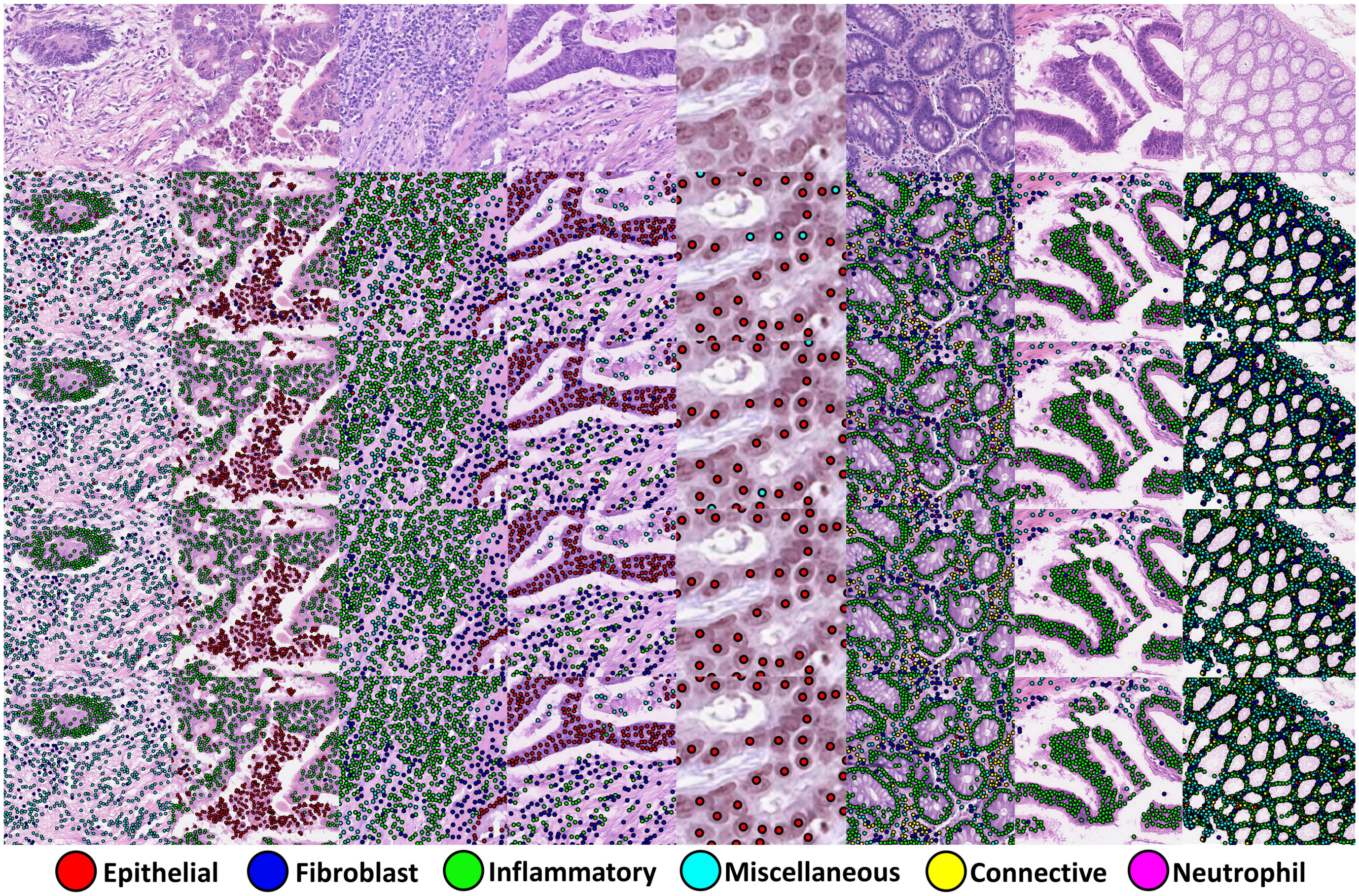

Fig. 6. Qualitative evaluation of the proposed scheme across CRCHistoPhenotypes [4], CoNSeP [5], PanNuke [24], and Lizard [7] datasets. The first row shows the original scans, the second row shows the results of Hover-Net [5], the third row shows the results of NCD [10], the fourth row shows the results of the proposed scheme, and the fifth row shows the ground truth annotations. Please zoom in for better visualization of the recognized nuclei communities.

(such as Hover-Net [5]) so that the misclassification due to false or missed detection of the nuclei regions can be minimized.

VII. Conclusion

This paper proposes a novel neural graph refinement scheme that exploits the homogeneity between the cellular structures using graph representational learning to differentiate the heterogeneous and homogeneous nuclei patterns. The proposed scheme is highly configurable and can be plugged into the existing state-of-the-art models to enhance their capacity for robustly identifying the nuclei types. The proposed system has been validated across four different histopathological datasets containing a wide variety of nuclei structures where, across all datasets, the proposed scheme demonstrates a highly accurate recognition of nuclei communities within the histopathological landscape, as compared to the state-of-the-art methods. Furthermore, the featurization block within the proposed scheme can be coupled with any graph partitioning and clustering algorithm, and this opens up new doors for the proposed scheme to be applied in vast variety of applications involving graph decomposition and clustering approaches (see Sections VI-A.7 and VI-A.8 for more details). In the future, we envisage validating the proposed method in clinical practice to effectively and efficiently extract the nuclei communities to grade cancerous progression. We also plan to test the proposed scheme toward recognizing different types of clinical biomarkers to measure the severity of the cancerous progression promptly. In addition to this, we also aim at exploring the proposed scheme in different application that are based on graph partitioning and clustering approaches.

References

[1] A. Das, M. S. Nair, and S. D. Peter, "Sparse representation over learned dictionaries on the Riemannian manifold for automated grading of nuclear pleomorphism in breast cancer," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1248–1260, Mar. 2019.

[2] S. Javed, A. Mahmood, N. Werghi, K. Benes, and N. Rajpoot, "Multiplex cellular communities in multi-gigapixel colorectal cancer histology images for tissue phenotyping," *IEEE Trans. Image Process.*, vol. 29, pp. 9204–9219, 2020.

[3] A. Paul and D. P. Mukherjee, "Mitosis detection for invasive breast cancer grading in histopathological images," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4041–4054, Nov. 2015.

[4] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1196–1206, May 2016.

[5] S. Graham et al., "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101563.

[6] W. Lu, M. Toss, E. Rakha, N. Rajpoot, and F. Minhas, "SlideGraph+: Whole slide image level graphs to predict HER2 status in breast cancer," 2021, *arXiv:2110.06042*.

[7] S. Graham et al., "Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 684–693.

[8] T. Hassan, M. Shafay, B. Hassan, M. U. Akram, A. ElBaz, and N. Werghi, "Knowledge distillation driven instance segmentation for grading prostate cancer," *Comput. Biol. Med.*, vol. 150, Nov. 2022, Art. no. 106124.

[9] Y. Zhou, S. Graham, N. A. Koohbanani, M. Shaban, P.-A. Heng, and N. Rajpoot, "CGC-Net: Cell graph convolutional network for grading of colorectal cancer histology images," in *Proc. IEEE ICCV-W*, Oct. 2019, pp. 388–398.

[10] T. Hassan, S. Javed, A. Mahmood, T. Qaiser, N. Werghi, and N. Rajpoot, "Nucleus classification in histology images using message passing network," *Med. Image Anal.*, vol. 79, Jul. 2022, Art. no. 102480.

[11] I. Pacal, D. Karaboga, A. Basturk, B. Akay, and U. Nalbantoglu, "A comprehensive review of deep learning in colon cancer," *Comput. Biol. Med.*, vol. 126, Nov. 2020, Art. no. 104003.

[12] H. Sharma et al., "A multi-resolution approach for combining visual information using nuclei segmentation and classification in histopathological images," in *Proc. 10th Int. Conf. Comput. Vis. Theory Appl.*, 2015, pp. 37–46.

[13] K. Nguyen, A. K. Jain, and B. Sabata, "Prostate cancer detection: Fusion of cytological and textural features," *J. Pathol. Informat.*, vol. 2, no. 2, p. 3, Jan. 2012.

[14] Y. Yuan et al., "Quantitative image analysis of cellular heterogeneity in breast tumors complements genomic profiling," *Sci. Transl. Med.*, vol. 4, no. 157, Oct. 2012.

[15] S. E. A. Raza et al., "Micro-Net: A unified model for segmentation of various objects in microscopy images," *Med. Image Anal.*, vol. 52, pp. 160–173, Feb. 2019.

[16] S. Abousamra et al., "Multi-class cell detection using spatial context representation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3985–3994.

[17] H. Kim et al., "Deep learning-based histopathological segmentation for whole slide images of colorectal cancer in a compressed domain," *Sci. Rep.*, vol. 11, no. 1, p. 22520, Nov. 2021.

[18] S. Tripathi and S. K. Singh, "Cell nuclei classification in histopathological images using hybrid $\mathrm{O_L}$ ConvNet," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 1s, pp. 1–22, Jan. 2020.

[19] L. Hou et al., "Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images," *Pattern Recognit.*, vol. 86, pp. 188–200, Feb. 2019.

[20] R. Ramirez et al., "Classification of cancer types using graph convolutional neural networks," *Frontiers Phys.*, vol. 8, pp. 1–14, Jun. 2020.

[21] L. P. Chew, "Constrained Delaunay triangulations," in *Algorithmica*, vol. 4. Berlin, Germany: Springer, Jun. 1989, pp. 97–108.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[24] J. Gamper et al., "PanNuke dataset extension, insights and baselines," 2020, *arXiv:2003.10778*.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[26] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, pp. 83–97, Mar. 1955.

[27] Q.-L. Zhang and Y.-B. Yang, "ResT: An efficient transformer for visual recognition," in *Proc. NIPS*, 2021, pp. 1–11.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[30] Y. Li, T. Yao, Y. Pan, and T. Mei, "Contextual transformer networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1489–1500, Feb. 2023.

[31] W. Tang, F. He, Y. Liu, and Y. Duan, "MATR: Multimodal medical image fusion via multiscale adaptive transformer," *IEEE Trans. Image Process.*, vol. 31, pp. 5134–5149, 2022.

[32] W. Tang, F. He, and Y. Liu, "TCCFusion: An infrared and visible image fusion method based on transformer and cross correlation," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109295.

[33] J. Zhu, M. Shao, C. Xia, H. Pan, and S. Xia, "Adversarial attacks on kinship verification using transformer," in *Proc. 16th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Dec. 2021, pp. 1–8.

[34] Y. Yu, Q. Yang, and L. Shaohan, "An improved faster R-CNN for colorectal cancer cell detection," in *Proc. IEEE ICECE*, Dec. 2020, pp. 186–190.

[35] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 731–737, Aug. 2000.

[36] Y. Su, Y. Bai, B. Zhang, Z. Zhang, and W. Wang, "HAT-Net: A hierarchical transformer graph neural network for grading of colorectal cancer histology images," in *Proc. Brit. Mach. Vis. Conf.*, 2021, pp. 1–15.

[37] W. Lou et al., "Structure embedded nucleus classification for histopathology images," 2023, *arXiv:2302.11416*.

[38] H. Zhu, L. He, M. Leeke, and R. Mao, "WolfGraph: The edge-centric graph processing on GPU," *Future Gener. Comput. Syst.*, vol. 111, pp. 552–569, Oct. 2020.