APSIPA Transactions on Signal and Information Processing, 2025, 14, e9

Original Paper

Two-stage Pipeline for Automated Cell Segmentation: Integrating Semantic and Instance Learning

Thanh-Ha Do[1*], Hoang Minh-Huong Dang[2], Thanh-Lam Tran[2] and Van-De Nguyen[3]

[1] *Posts and Telecommunications Institute of Technology, Hanoi, Vietnam*
[2] *VNU University of Science, Hanoi, Vietnam*
[3] *The 108 Military Central Hospital, Hanoi, Vietnam*

ABSTRACT

Accurate segmentation of individual thyroid cells is a prerequisite for cell feature analysis and reliable cancer staging classification. However, Diff-Quick stained cytology images present significant challenges: frequent misclassification of malignant cells and erythrocytes and substantial cell overlap hindering boundary detection. To address these issues, we propose a novel two-stage pipeline. This approach enhanced the efficiency-optimized nnU-Net v2 for rapid foreground-background separation, enabling efficient instance segmentation of overlapping cells and reducing erythrocyte misclassification. The results evaluated on a dataset with multiple thyroid cancer stages show that our method reduced erythrocyte false positives and improved accuracy over the best post-processed baseline while cutting inference time. These findings demonstrate the practical utility of our pipeline for automated Diff-Quick thyroid cytology image segmentation within real-world clinical workflows.

[*] Corresponding author: dothanhha@ptit.edu.vn. This research was funded by research project QG.23.71 of Vietnam National University, Hanoi.

Received 23 May 2025; revised 28 May 2025; accepted 03 June 2025
ISSN 2048-7703; DOI 10.1561/116.20250033

Keywords: Diff-quick cell images, microscopic, cell segmentation, automatic, deep learning

1 Introduction

DiffQuick is a rapid, cost-effective staining protocol that colors nuclei and cytoplasm differently. It is indispensable for routine cytological screening, particularly in thyroid cancer diagnosis. However, its very practicality introduces a set of imageanalysis challenges. First, the stain often produces uneven chromatic contrast and hue variation, hampering classical intensity-based algorithms. Second, aspirates from thyroid nodules typically contain densely packed, partially overlapping cells whose nuclei share difficult-to-distinguish borders.

These factors make automated segmentation and subsequent classification far more difficult than in curated benchmark datasets. Consequently, there is a pressing need for advanced learningbased approaches that can (i) disentangle overlapping cellular structures, (ii) remain robust to color and illumination variability, and (iii) explicitly differentiate thyroid cells from visually similar red blood cells while still operating within the inferencetime constraints of busy clinical environments.

Effective cell segmentation is essential for extracting cellular features that aid in cancer diagnosis. Traditionally, this problem has been addressed using classical image analysis techniques [34, 21, 33, 19, 18]. These methods rely on the inherent intensity or other properties within medical images to distinguish cells, frequently thresholding. Thresholding classifies pixels as belonging to a cell or background based on whether their intensity exceeds a defined value. This threshold can be a single value applied globally [17] or can vary locally to account for image heterogeneity.

In addition, watershed segmentation [23] has been utilized for cell segmentation in microscopic images. Initially, the process involves identifying pixels that delineate the foreground and background layers of the image. Subsequently, these delineated areas are expanded by incrementally elevating the image intensity level. The regions surrounding the marked pixels are called watersheds, where each pixel value represents the terrain. Boundaries between adjacent basins are established by lines of maximum elevation, known as watershed lines.

Based on the concepts of classical segmentation techniques, such as thresholding and watershed methods, clustering is another critical approach in cell segmentation. Clustering utilizes various algorithms [20, 26, 25] to group pixels with similar attributes. Among these, K-Means [12] is the most prevalent. Alternative clustering algorithms, such as Fuzzy C-Means [3] and Mean Shift [4], have also been successfully applied in cell segmentation tasks.

The active contour method and its variations play a significant role in automatic cell image segmentation [1, 6, 31, 15]. In this method, border points are crucial as they help establish the minimum energy level necessary for defining nuclei boundaries. The energy function, integral to this model, is meticulously designed to penalize both discontinuities in the shape of the curve and discrepancies in gray levels along the contour and ensures that the segmentation accurately follows the natural boundaries of the cells.

Classical image analysis techniques are often used for simplicity, including thresholding, watershed segmentation, and clustering. However, their effectiveness is limited in noise or high-intensity variations images. Furthermore, they often struggle with clustered or irregularly shaped cells. While active contour models and graph-based segmentation can address some of these limitations, they typically require substantial computational resources, especially for images with densely distributed cell nuclei. Consequently, deep learning models have recently garnered significant attention in medical image analysis. Within this domain, two primary deep learning-based approaches have emerged [8, 2]: (1) employing deep learning models like U-Net [24] in conjunction with post-processing for individual cell segmentation, and (2) utilizing dedicated segmentation models based on object detection principles, such as Mask R-CNN [7, 11].

Deep learning models are developed for analyzing pathological images and have found widespread application by various research organizations [14, 30]. For instance, Sharma *et al.* [27] developed a system capable of classifying stomach cancer from whole-slide images, while Korbar *et al.* [13] devised a method for classifying colorectal polyps on whole-slide images. Additionally, Elman Neural Networks (ENNs) [5] have been employed in constructing computer-aided thyroid cytology diagnosis methods. In recent developments, tools such as Nuclei AIzer have demonstrated the feasibility of fully automated pipelines by integrating a Mask R-CNN backbone enhanced with style transfer techniques to improve generalization across datasets with varied staining styles and using the U-Net model to refine the segmentation masks. Despite these advancements, deep learning models remain underutilized in cytological diagnosis, including thyroid cytology, warranting further exploration and development in this domain.

During our experimental study, we observed the advantages of employing U-Net-based models for semantic segmentation and Mask R-CNN for instance-level segmentation. We implemented a series of data post-processing and enhancement techniques to systematically evaluate these models to improve segmentation accuracy on real-world thyroid cytology images. The results demonstrate that such accuracy improvements often come at the cost of increased inference time, posing challenges for practical deployment in time-sensitive clinical settings.

This trade-off between precision and efficiency motivated us to explore alternative architectures and optimization strategies. As a result, we developed the integrated pipeline presented in this work, which balances segmentation performance and computational speed by combining the strengths of state-of-the-art deep learning models with an efficient workflow tailored for Diff-Quick-stained cytology images.

In this paper, we propose an automated segmentation flow to address the specific challenges of Diff-Quick staining cytometry imaging. Our approach leverages the strengths of deep learning by integrating two state-of-the-art architectures: nnU-Net v2 [10] for semantic segmentation and Cellpose for instance-level segmentation. The nnU-Net v2 model is employed to distinguish the cellular foreground from the background with minimal manual configuration, adapting dynamically to the dataset's properties. Meanwhile, Cellpose [28] is effective in resolving overlapping cell regions and detects individual cell instances, a model designed as a flexible API for experts in the field of medical image segmentation. To validate our approach, we conducted extensive experiments on a self-constructed separate dataset of thyroid cell images collected from the 108 Military Central Hospital in Vietnam. The evaluation demonstrates that our combined framework performs well in segmentation compared to classical baselines and single models, especially in densely packed and morphologically diverse regions.

The remainder of this paper is organized as follows. Section 2 reviews deep learning-based segmentation methods, focusing on the two models adopted in our framework: nnU-Net and Cellpose. Section 3 outlines the proposed segmentation workflow. Section 4 details the self-constructed dataset used for cell segmentation. Section 5 presents the experimental results and corresponding analysis. Finally, Section 6 concludes the paper and discusses potential directions for future research.

2 Deep Neural Networks for Cell Segmentation in Microscopic Imaging

Deep learning models have transformed digital pathology by facilitating the end-to-end learning of rich morphological features that classical pipelines cannot effectively achieve. This section reviews two kinds of segmentation architectures for cellular segmentation that are considered the baseline for developing and comparing our proposed models: the UNet and Mask R-CNN. The U-Net [24] is presented as a representative baseline for semantic segmentation, owing to its demonstrated effectiveness in medical image competitions and simplicity for rapid prototyping. For instance, segmentation, Mask R-CNN [7] is highlighted for its strong performance on COCO-style datasets and its broad adoption in biomedical research. Therefore, we revised these two influential models and their variants.

In addition, this section also describes two deep learning models chosen to develop in the proposed segmentation workflow, nnU-Net, and Cellposethe reason for selecting nnU-Net and Cellpose is based on the self-built experiment.

2.1 The U-Net Model and its Variants

The UNet was proposed by Ronneberger *et al.* [24] and follows a symmetric *EncoderDecoder* design with long skip connections that ferry highresolution features from the contracting to the expanding path, thus preserving localization while mitigating vanishinggradient effects.

The U-Net backbone consists of repeated encoder and decoder stages. Each encoder stage applies two 3×3 convolutions and ReLU, followed by 2×2 maxpooling for downsampling. The decoder mirrors this layout, replacing maxpooling with 2×2 transposed convolutions to upsample and refine the feature maps. During inference, overlappingtile prediction is employed to reduce boundary artifacts, and extensive data augmentation (elastic deformation, rotation, intensity jitter) improves generalizationan essential property for our DiffQuick thyroid dataset, where annotated samples are scarce and stain variability is high.

Since its introduction in 2015, a family of extensions has emerged, including UNet++ [35], which nests dense skip connections to reduce the semantic gap between encoder and decoder feature maps, and nnUNet v2 [10], which automates the hyperparameter tuning process, eliminating the need for manual adjustments. This pipeline dynamically adapts crucial parameters like patch size, loss function, and data augmentation policy based on the characteristics of each specific dataset.

2.2 nnU-Net Models

The nnU-Net model is a deep learning framework built upon the widely adopted U-Net architecture. However, it significantly extends it by introducing a fully automated configuration pipeline tailored for biomedical image segmentation.

The nnU-Net pipeline includes four core components:

1. Preprocessing: In the preprocessing phase, nnU-Net analyzes features of the input data, such as voxel spacing, to determine the image's spatial dimensionality (2D, 3D, or 4D), spatial resolution, and pixel intensity distribution. Based on these features, the model selects a preprocessing strategy to ensure consistency and optimization for the training process.

   Specifically, nnU-Net applies z-score normalization to standardize the pixel intensity distribution. Each pixel x is transformed to z according to the formula in Equation 1.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Where μ and σ denote the mean and standard deviation of the image, respectively.

For CT images, the model uses a percentile cutoff to remove extreme intensity outliers, followed by global statistical normalization. Additionally, if the data includes 3D or 4D volumes, the images are resampled to standardized voxel spacing using spline or nearest-neighbor interpolation methods to ensure spatial resolution uniformity. Label maps are also converted to one-hot encoding and interpolated appropriately, which is particularly important for anisotropic volumes to preserve accurate boundaries in three-dimensional space.

However, in this paper, all data are 2D images with uniform resolution. Therefore, nnU-Net only applies z-score normalization and does not require spatial resampling or other operations related to volumetric data.

2. Network Architecture Configuration: After preprocessing, nnU-Net configures the network architecture based on key characteristics of the dataset and hardware information. The average image shape and voxel spacing are analyzed to determine the appropriate number of resolution levels (i.e., the U-Net depth), patch size, and overall network structure.

   The initial patch size is set to match the mean shape of the input image to maximize contextual coverage. The nnU-Net model then estimates the GPU memory required for training and, if necessary, gradually reduces the patch size to match the available hardware. Once the patch size is finalized, the batch size is determined to optimize GPU memory while maintaining a minimum batch size of two for stable training (see Equation 2).

$$\text{VRAM}_{\text{est}} \approx B \cdot \sum_{l=1}^{L} C_l \cdot H_l \cdot W_l \cdot D_l \tag{2}$$

   Where B is the batch size, and C_l, H_l, W_l, D_l are the number of channels and spatial dimensions (height, width, depth) of the feature maps at layer l. This heuristic trade-off between patch size and batch size enables efficient dataset training.

   All network configurations generated by nnU-Net are based on the original U-Net regularization architecture [24] and its 3D extension. The nnU-Net model adjusts some parameters and significantly improves the efficiency and stability of the training process. Specifically, to accommodate large input patches while remaining within memory constraints, the nnU-Net model typically trains with mini-batch sizes. Since batch normalization is sensitive to mini-batch sizes and can degrade performance under such conditions, nnU-Net replaces it with instance normalization across all layers. Furthermore, the ReLU activations are replaced with leaky ReLUs (negative gradient = 0.01) to maintain gradient flow in low activation regions.

The architecture follows the classic encoder-decoder scheme with skip connections and deep supervision. Two convolutional blocks are applied in the encoder and decoder at each resolution level. Each block consists of a convolution layer, instance normalization, and leaky ReLU activation. Downsampling uses strided convolutions, while upsampling uses transposed convolutions. To balance computational cost, the network initializes with 32 feature maps and doubles this number at each downsampling stage while halving during upsampling. The total number of feature maps is further capped at 320 for 3D U-Nets and 512 for 2D variants to constrain model size.

3. Training Strategy: The training process in nnU-Net is automatically configured to optimize convergence and generalization. The framework employs a composite loss function that combines Dice and cross-entropy loss, addressing class imbalance and per-pixel classification accuracy. The total loss combines the soft Dice loss and the pixel-wise cross-entropy loss as in Equation 3.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{CE}} \tag{3}$$

where the soft Dice loss is defined in Equation 4.

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i + \epsilon} \tag{4}$$

The cross-entropy loss is defined as in Equation 5.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} g_{i,c}\log(p_{i,c}) \tag{5}$$

Here, p_i and g_i denote the predicted and ground-truth probabilities for pixel i, and C is the number of classes. ϵ is a small constant added for numerical stability. The Dice encourages overlap between predicted and ground truth regions, while the cross-entropy penalizes pixel-wise classification errors. This hybrid loss is particularly well-suited for biomedical segmentation tasks, which often suffer from class imbalance.

Optimization is carried out using stochastic gradient descent (SGD) with Nesterov momentum [29], a strategy known to improve convergence speed and stability in deep networks by incorporating a look-ahead gradient step.

4. Postprocessing: In the final stage, nnU-Net performs resampling of the predicted segmentation masks to match the original image resolution and spacing.

2.3 Instance Segmentation: The Mask R-CNN and its Variants

Instance segmentation builds upon object detection by generating a pixel-level binary mask for each identified object. Mask R-CNN, introduced by He *et al.* [7], is a sophisticated two-stage framework that enhances Faster R-CNN [22] with an additional branch dedicated to predicting segmentation masks in parallel with object detection. Its main components include:

- Backbone and Feature Pyramid Network (FPN): A deep convolutional neural network (CNN), such as ResNet-50, combined with a Feature Pyramid Network, extracts multi-scale feature maps from the input image.

- Region Proposal Network (RPN): This network generates class-agnostic Region of Interests (RoIs) potential object locations. Positive anchor boxes have an Intersection over Union (IoU) greater than 0.7 with any ground-truth bounding box, while negative anchors have an IoU less than 0.3.

- RoIAlign: This module replaces the previous RoIPool operation with bilinear interpolation [32] to precisely align the RoIs with the feature maps at the pixel level. This accurate alignment is particularly crucial for microscopy images where fine details matter.

- Multi-task heads: Each fixed-size RoI is then processed by two parallel heads: (i) a classifier that predicts the object's class and a bounding-box regressor that refines the object's spatial extent, and (ii) a KŒmŒm mask predictor that outputs a binary segmentation mask for each of the K classes using a sigmoid activation function.

2.4 Cellpose Model

Unlike conventional approaches that perform per-pixel classification, Cellpose introduces a novel formulation of instance segmentation as a vector flow regression task (see Figure 1).

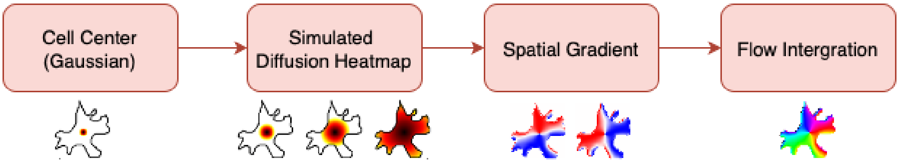

Figure 1: Calculate flow vector in cellpose. (Note: adapted from [28]).

2.4.1 Vector Flow-based Representation of Object Masks

The core idea is to represent each object as a spatial flow field that guides pixels toward the centroid of the corresponding cell.

During training, a diffusion process is simulated starting from the object center, producing a scalar heat map for each ground-truth mask. The gradient of this heat map concerning spatial coordinates defines a vector field $\vec{F}(x, y)$ that points inward toward the object center (see Equation 6)

$$\vec{F}(x,y) = -\nabla h(x,y) \tag{6}$$

Where $h(x, y)$ is the diffusion-based scalar field generated by solving Equation 7.

$$\frac{\partial h}{\partial t} = \Delta h, \quad h(x,y,t=0) = \delta(x - x_c, y - y_c) \tag{7}$$

Here, Δ denotes the Laplacian operator, which represents the sum of second-order spatial derivatives of the scalar field $h(x, y)$, capturing the rate at which the field diffuses across space. Formally, in two dimensions, it is defined as in Equation 8.

$$\Delta h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} \tag{8}$$

Moreover, δ is a Dirac delta function centered at the object's centroid (x_c, y_c). In practice, the Dirac delta function $\delta(x-x_c, y-y_c)$ used to initialize the heat map is approximated by a narrow Gaussian (see Equation 9).

$$\delta(x,y) \approx \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x_c)^2 + (y-y_c)^2}{2\sigma^2}\right) \tag{9}$$

The resulting flow field encodes object shape implicitly and is used as the training target for the neural network.

At inference time, Cellpose predicts a pair of spatial flow components $(\vec{F}_x, \vec{F}_y)$ along with a probability mask. To reconstruct individual cell instances, the predicted flow field is integrated via gradient descent, where each pixel “flows” along its vector direction until convergence. Pixels that converge on the exact center point are grouped into the same object mask.

This flow-based formulation allows Cellpose to segment various cellular structures, including irregular and non-convex shapes, without explicit boundary detection. The network architecture is based on a modified U-Net with residual blocks and a global style embedding vector that helps adapt to image-specific characteristics.

2.4.2 Training Data and Supervision Strategy

To better illustrate the internal workings of Cellpose, Figure 2 highlights key components of its architecture and prediction mechanism. The model reformulates instance segmentation as a vector field regression problem, where each pixel is guided toward the center of its corresponding cell. Below, we present the training objective and architectural details that enable this approach.

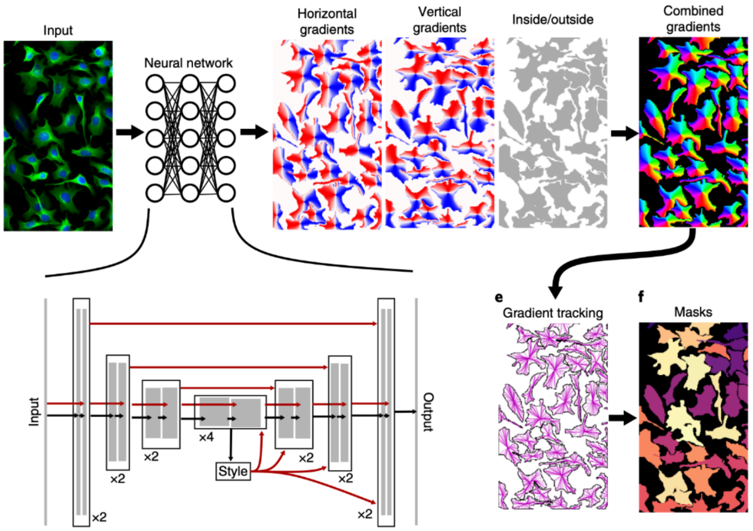

Figure 2: **Model Architecture**. The model is trained to output two directional gradient maps (horizontal and vertical) and a segmentation probability map, forming a vector field representing the direction of cell boundaries. The network uses a U-Net architecture, incorporating both encoderdecoder paths with skip connections and global style features, allowing it to generalize across diverse image styles and resolutions [28].

The U-Net backbone employed in Cellpose differs from the standard implementation in several key ways to handle better the morphological variability and staining inconsistencies typical of cytological images. First, the architecture incorporates residual connections within each block, improving gradient flow and stability during training. The network is also deepened compared to the original U-Net to enhance its capacity for feature extraction.

Instead of using feature concatenation between the encoder and decoder paths, Cellpose employs element-wise summation to reduce model complexity. Furthermore, Cellpose integrates a global style representation extracted by average pooling in the network bottleneck to accommodate image-specific staining styles. This style vector is injected into all decoder stages, allowing the model to dynamically adjust to variations in staining intensity and visual appearance across different samples.

Cellpose was trained on a large-scale, heterogeneous dataset of over 70,000 manually annotated objects to achieve instance segmentation across various image types. The dataset encompasses a broad spectrum of microscopy modalities, including fluorescence, phase contrast, brightfield imaging, and diverse staining techniques. It includes nuclear and whole-cell masks, covering various cell morphologies, densities, and textures.

The training supervision is based on the flow vector field $(\vec{F}_x, \vec{F}_y)$ and binary mask prediction. This dual-objective learning strategy allows Cellpose to simultaneously learn semantic object localization and precise instance-level shape encoding via vector flow fields.

The flow loss is defined as the mean squared error (MSE) between the predicted and ground-truth flow vectors (see Equation 10).

$$\mathcal{L}_{\text{flow}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \vec{F}_{\text{pred}}(x_i, y_i) - \vec{F}_{\text{gt}}(x_i, y_i) \right\|^2 \tag{10}$$

In parallel, a binary cross-entropy loss is applied to supervise the object probability map (see Equation 11).

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \left[g_i \log p_i + (1 - g_i) \log(1 - p_i) \right] \tag{11}$$

where $g_i \in \{0, 1\}$ denotes the ground-truth label and p_i the predicted foreground probability for pixel i.

The total loss is a weighted sum of the two components as indicated in Equation 12.

$$\mathcal{L}_{\text{total}} = \lambda_{\text{flow}} \cdot \mathcal{L}_{\text{flow}} + \lambda_{\text{BCE}} \cdot \mathcal{L}_{\text{BCE}} \tag{12}$$

3 Single Model vs. Combined Pipeline: Proposed Deep Learning Strategies for Cell Segmentation

This section presents a series of targeted experiments to explore and validate the core ideas proposed in this work.

3.1 Proposed Single Model for Cell Segmentation

The proposed single model is illustrated in Figure 3.

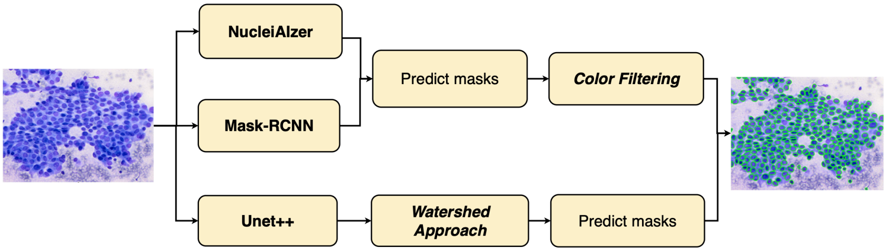

Figure 3: Proposed single model for cell segmentation.

We first cropped the input image into small overlapping patches for the instance segmentation branch (the Mask R-CNN model and the NucleiAIzer method). This approach ensures accurate segmentation at the cellular level by preserving image resolution and avoiding the downscaling that often degrades detail, especially in pipelines like Mask R-CNN, which typically resize inputs to fixed dimensions. After segmentation, the predicted masks for each patch were stitched back to the original image size, followed by a color-based filtering step to retain relevant foreground regions.

In contrast, for the semantic segmentation branch (U-Net++), we directly input the whole image into the model without patch cropping. The resulting binary prediction mask was refined using a watershed algorithm to delineate boundaries between individual cell regions.

The following describes the additional processing steps applied to each model branch to better localize cell regions and map contours into separate individual cells.

3.1.1 Combine Mask-RCNN and NucleiAIzer with Color-based Filtering for Red Blood Cell Suppression

We observed that red blood cells and thyroid cells usually have different color intensities. This makes thresholding methods like Otsu [16] a reasonable choice for separating them. In our method, we first calculate the average color intensity of each segmented region. Then, we apply Otsus thresholding to automatically choose an optimal threshold that separates darker and lighter regions. Otsus method selects the threshold t^* that maximizes the between-class variance (see Equation 13).

$$t^* = \arg\max_t \left[\omega_0(t) \cdot \omega_1(t) \cdot (\mu_0(t) - \mu_1(t))^2 \right] \tag{13}$$

Where $\omega_0(t)$ and $\omega_1(t)$ are the probabilities of the two classes separated by threshold t, and $\mu_0(t)$, $\mu_1(t)$ are their corresponding mean intensities.

This filtering step was initially used to reduce false positives caused by red blood cells. However, the separation became less reliable in some instances, particularly when erythrocytes appeared unusually faint or shared similar intensity with thyroid nuclei.

3.1.2 Combine U-Net++ Model to Watershed Post-processing

Predicted probability maps from U-Net++ were first binarized to identify cell regions. To handle overlapping structures, we applied a classical watershed-based post-processing strategy as shown in Figure 4 Details of the post-processing stages are as follows:

- Remove noise and small holes in the cell region from the binary image using morphological opening operations.
- Perform image relaxation, marking areas that contain the background and the foreground image layer in image A.
- Apply distance transform to calculate the distance from each pixel to the nearest 0-valued pixel.
- Perform thresholding on the distance-transformed image, resulting in image B, representing the foreground image layer.
- Subtract image B from image A to obtain image C, marking areas uncertain to belong to either the background or the foreground image layer.
- Find connected components in image B, marking uncertain areas within these connected components as 0 to obtain border markings around the areas, known as watersheds.
- Use the watershed principle to segment using the previously marked boundaries, taking the watershed boundaries as cell borders to obtain the final segmented image, delineating each cell.

3.1.3 Combine nnU-Net v2 to Post-processing Strategy using Centroid-guided Region Growing

Before developing the final approach using nnU-Net combined with Cellpose, we also explored another post-processing method to address the cell segmentation task. In this direction, we propose the generation of semantic segmentation masks from nnU-Net and aim to separate overlapping cells by identifying

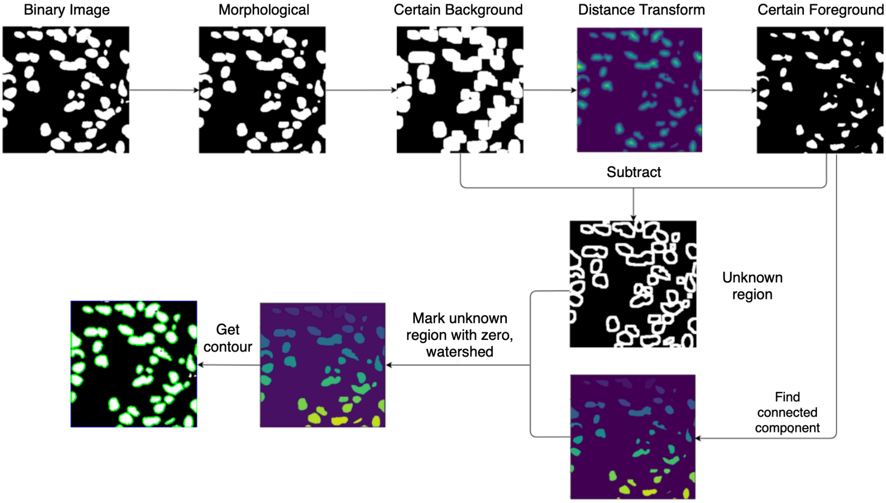

Figure 4: Watershed-based post-processing pipeline applied to semantic segmentation results from U-Net++.

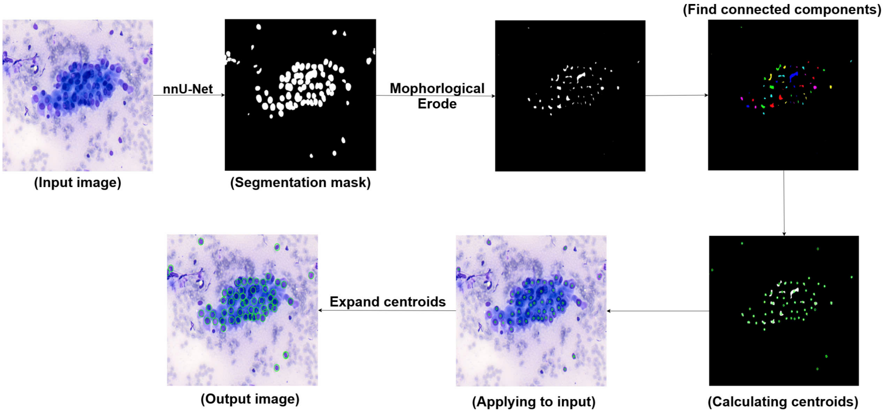

Figure 5: Alternative post-processing strategy: centroid-guided region growing.

centroids and expanding them to delineate boundaries. The detailed steps are indicated in Figure 5.

- Semantic Segmentation Mask Generation: The process began with the binary mask from nnU-Net, in which white pixels represented cells. Due to cell overlap, connected regions often had to be separated.

- Morphological Operations for Cell Separation: Erosion with a circular kernel ($r = 2$) was applied to shrink regions and create gaps between overlapping cells. This made individual components more separable.
- Finding Connected Components and Centroids: After erosion, connected components were identified, and centroids were calculated using image moments:

  $$x_c = \frac{M_{10}}{M_{00}}, \quad y_c = \frac{M_{01}}{M_{00}} \tag{14}$$

  where $M_{pq} = \sum_x \sum_y x^p y^q I(x, y)$.
- Region Expansion: Each centroid was mapped back and expanded outward to recover the original cell region, mimicking the reversed erosion process.

By experiment (see more detail in Section 4), we recognize that this approach suffered from several key limitations:

1. It was ineffective in cases of strong overlap, leading to shape distortion or missed detections.
2. The number of erosion steps varied across images, requiring manual tuning.
3. The lack of automation made it impractical for large-scale deployment.

In summary, these limitations motivated the development of a more streamlined and fully automated solution, which we detail in Section 3.2 describing our proposed combined pipeline.

3.2 Proposed Combined Pipeline for Cell Segmentation

Motivated by these shortcomings, we propose a two-stage segmentation framework tailored explicitly for Diff-Quick-stained thyroid cytology images. Our new approach leverages a self-configuring semantic segmentation framework (nnU-Net) to robustly distinguish cell regions from the background, followed by the Cellpose instance segmentation model to accurately and efficiently delineate individual cell boundaries, even in densely packed and overlapping scenarios.

This section details the two complementary components of the proposed pipeline: *nnU-Net* for semantic segmentation (Section 2.2); and *Cellpose* for instance segmentation (Section 2.4).

Table 1: Segmentation performance of U Net and its variants on the DiffQuick test set.

Model	**Dice** ↑	**Inference time** (s) ↓
U Net (baseline)	0.782	45
U Net++	0.826	60
nnU Netv2	**0.842**	**43**

The reason to choose the nnU-Netv2 instead of Unet is that Dice scores (see Table 1) and average inference time on our DiffQuick thyroid cytology test set (details of the evaluation protocol in Section 5).

The progressive improvements validate the benefit of architectural enhancements, while nnUNet v2 attains the best tradeoff between accuracy and speed, motivating its use in our subsequent twostage pipeline.

Table 2 reports the performance of Mask R-CNN and its variant on our DiffQuick preliminary test set. The preliminary dataset was a test subset designed to assess segmentation performance under relatively clean conditions, particularly focusing on cases with overlapping cells. Mask R-CNN and its variants performed well on this subset in delineating individual cell instances. We chose Cellpose because although the Mask RCNN variants excel at delineating precise boundaries for every cell, their inference time is still prohibitively long for routine clinical use.

Table 2: Mask R-CNN on DiffQuick test set and inference time.

Configuration	**Dice Score** ↑	**Time** (s) ↓
Mask R–CNN	0.736	198
NucleiAIzer [9]	0.81	195
Cellpose [28]	**0.86**	**15**

Furthermore, we extended the dataset to include more challenging conditions, such as varying staining quality, dense cellular arrangements, and overlapping instances. Under these harsher scenarios, the integration of both models demonstrated a balanced trade-off between segmentation accuracy and efficiency, detailed in Section 5.

4 Self-constructed Cell Segmentation Dataset

The data was provided with the patient's permission following the project from the Vietnam National University, and the patient's personal information was anonymized entirely. We collected from many typical stages of thyroid cancer on the Bethesda scale. We designed a tool for doctors to perform image pushing and contour labeling operations on images. A team of five expert doctors performed cell contouring.

Our annotated dataset currently contains around 300 Diff-Quick-stained cytology images captured at 20 × magnification, each with a 1024 × 768 pixels resolution. According to the Bethesda system, these images are categorized into three disease stages: B4, B5, and B6. Each image includes approximately 80 to 271 cells, providing a sufficiently diverse and representative dataset for training and evaluating the proposed segmentation methods. In total, we labeled 20,374 cells.

Examples of the input data are shown in Figure 6. Most of the images in the dataset were captured at a zoom level of 20 micrometers, providing detailed cellular structures suitable for segmentation. To generate high-quality training labels, we developed a custom annotation tool allowing expert pathologists to outline individual cell contours manually. All annotations were verified for accuracy by the same medical professionals.

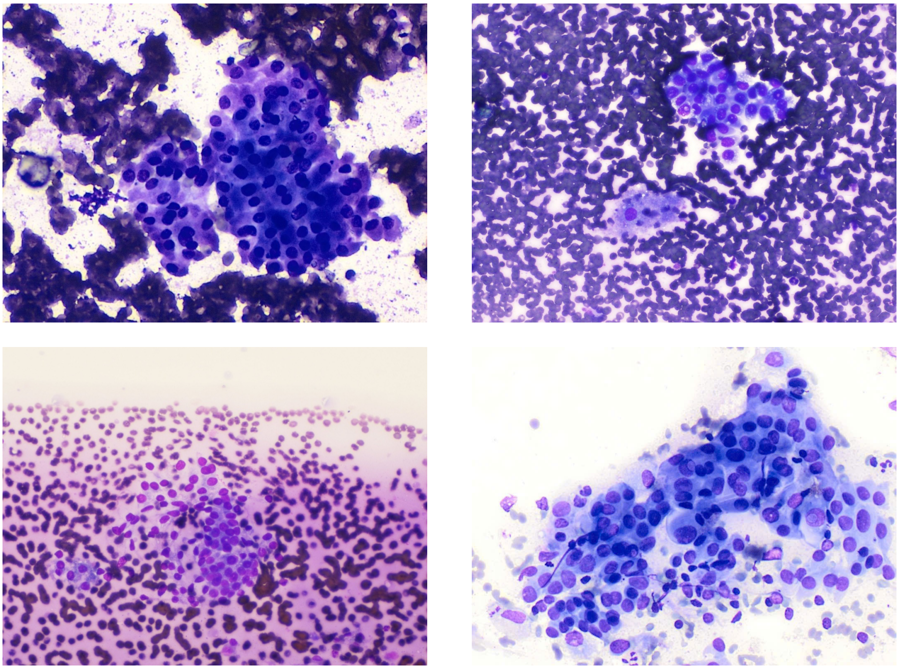

Figure 6: Thyroid cancer cell images from the dataset.

5 Experiment Result

5.1 Database

The dataset comprises 272 thyroid cancer cell images, each with a resolution of 1024 × 768 × 3. The data is divided into training, validation, and testing sets as shown in Table 3. Details regarding the data augmentation process applied during training is described in Section 5.2.

Table 3: Overview of the dataset.

	Training	**Validation**	**Testing**	**Total**
Images	120	30	122	277
Cells	8990	2207	9177	20,374

5.2 Parameters

5.2.1 Data Augmentation

The dataset initially contains 272 thyroid cancer cell images, which is relatively small for effectively training a deep learning model. To improve the learning capability and accuracy of the nnU-Net model, we applied several data augmentation techniques to enhance the diversity of the training and validation sets, which consist of 150 images. Specifically, the augmentations are applied dynamically during training with specific probabilities: each image, when passed through the training pipeline, has a chance to undergo each augmentation, which may be applied or skipped based on its defined probability. This way, in every epoch, the model "sees" different versions of the same original images, thereby increasing data diversity and improving the model's generalization during training.

1. Random flipping of images was applied with a probability of 50%.

2. Images were randomly rotated within a range of $[-180°, 180°]$ with a probability of 20%.

3. Random scaling was applied with a scaling factor in the range $[0.7, 1.4]$ with a probability of 20%.

4. Low resolution transformation was applied with a scaling factor in the range $[0.5, 1.0]$ with a probability of 25%.

5. Gamma transformation was applied with a gamma value in the range $[0.7, 1.5]$ with a probability of 30%.

6. Brightness transformation was applied with a scaling factor in the range $[0.75, 1.25]$ with a probability of 15%.

7. Contrast transformation was applied with a scaling factor in the range $[0.75, 1.25]$ with a probability of 15%.

8. Gaussian blur was added with a random standard deviation σ in the range $[0.5, 1.0]$ with a probability of 20%.

9. Gaussian noise was added with a noise variance in the range $[0, 0.1]$ with a probability of 10%.

5.2.2 Training Configuration

The nnU-Net framework was configured with a 2D U-Net architecture tailored to our dataset, consisting of an encoder with eight pooling layers using max-pooling and convolution operations, followed by a decoder with eight upsampling layers to recover the spatial dimensions. Each layer in the encoder and decoder incorporated convolution operations, Instance Normalization, and the LeakyReLU activation function. During the preprocessing phase, nnU-Net extracted the dataset fingerprint, determining a median image shape of 768×1024 (height $\times$ width). Consequently, the patch size was set to 768×1024, and the batch size was optimized to 4 based on a GPU memory consumption analysis, balancing memory usage and training efficiency.

The nnU-Net model was trained for 50 epochs using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 7×10^{-3}, increased from the default value of 5×10^{-3} to improve convergence speed. A weight decay of 1×10^{-4} was applied to prevent overfitting. The foreground oversampling percentage was set to 0.33 to address the class imbalance, ensuring that 33% of the sampled patches contained foreground regions. The training process included 250 iterations per epoch, while validation was performed with 50 iterations per epoch. The loss function combined Dice Loss and Cross-Entropy Loss to focus on cancer cell regions.

5.3 Measurement

We evaluated the segmentation performance using multiple metrics to assess the overlap between predicted and ground truth segmentation masks across the B4, B5, and B6 stages. We employed the Dice Coefficient and Intersection over Union (IoU) metrics for semantic segmentation, which measure pixel-wise overlap between the predicted and ground truth masks. For instance, in segmentation, we utilized precision to evaluate the accuracy of individual instance predictions.

Dice Coefficient quantifies the overlap between predicted and ground truth masks, emphasizing the balance between true positives and errors. It is defined as in Equation 15.

$$\text{Dice} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{15}$$

In which (TP) (True Positives) represents the number of pixels that were correctly predicted as belonging to a cell and are indeed cell pixels. (FP) (False Positives) miscount the pixels predicted as cells, but part of the background. Conversely, (FN) (False Negatives) signifies the number of pixels that are genuinely cells but were mistakenly predicted as background.

Intersection over Union (IoU) measures the intersection ratio to the union of the predicted and ground truth masks, providing a robust metric for pixel-wise segmentation accuracy.

Precision for instance segmentation evaluates the proportion of predicted instances that are correctly identified as true instances. We adopted a best-match approach, where each predicted instance is paired with the ground truth instance yielding the highest IoU, requiring an IoU score of at least 0.7 to ensure high-quality matches. This approach was chosen for instance segmentation because it focuses on the accuracy of instance detection. It is critical to evaluate the model's ability to distinguish individual cells in dense or overlapping regions, as often encountered in images of thyroid cancer cells. Unlike Dice and IoU, which assess pixel-wise overlap, Precision directly measures the correctness of instance predictions, making it suitable for quantifying the model's performance in identifying distinct cell instances.

5.4 Result

The final evaluation was conducted on a set of tests of 122 thyroid cytology images, covering various stages of the disease (32 B4 images, 50 B5 images, and 40 B6 images). We first assessed the semantic and instance segmentation results of several single-model architectures to evaluate the baseline performance.

For semantic segmentation, we compared U-Net, U-Net++, and nnU-Net v2. As shown in Table 4, nnU-Net v2 achieved the highest Dice and IoU scores.

Table 4: Result of single model for semantic cell segmentation.

Model	**Dice Score**	**IoU Score**	**Inference time** (s)
Unet	0.76	0.62	45
Unet++	0.78	0.62	60
Enhance Unet++	0.81	0.65	65
nnUnetv2	**0.89**	**0.82**	**43**

For instance segmentation, we compared three representative models: Mask R-CNN, NucleAIzer, and Cellpose, as summarized in Table 5.

Table 5: Result of single model for instance cell segmentation.

Model	**Dice Score**	**Precision Score**	**Inference time** (s)
Mask-RCNN	0.70	0.51	198
NucleAIzer	**0.75**	**0.59**	195
Cellpose	**0.71**	**0.59**	**10**

The proposed two-stage pipeline outperforms all single-model baselines in accuracy and inference time. Table 6 shows that it achieves the best Dice, IoU, and Precision scores.

Table 6: Result of two-stage pipeline proposed.

Model	**Dice**	**IoU**	**Precision**	**Inference time** (s)
nnUnetv2+Cellpose	**0.92**	**0.86**	**0.87**	**60**

Figure 7 illustrates the results obtained by nnUnet and the nnUnet combined with Cellpose. As indicated in this figure, the nnU-Net model effectively separates cellular foreground and background; however, it does not work well if the cells overlap. The combined nnU-Net and Cellpose overcome this limitation, indicating adequate delineation of individual cells.

A notable highlight of the combined nnU-Net and Cellpose approach is its excellent performance and execution time, which is approximately 60 seconds per image. This is a significant improvement compared to the results of previous research. This efficiency in both performance and speed is the key motivation behind developing a new method, as presented in this paper, to address the time constraint while substantially enhancing segmentation performance.

6 Conclusion

This study firmly establishes that the proposed method has demonstrated exceptional effectiveness in Instance Segmentation of thyroid cancer cells, achieving outstanding accuracy and significantly reduced processing time compared to traditional approaches. The integration of the nnUNet model for semantic segmentation and the Cellpose model for instance segmentation has proven to be a transformative approach, enabling precise differentiation of individual cells within complex clusters, even in cases with irregular shapes and large nuclei. This breakthrough addresses the limitations of manual segmentation and

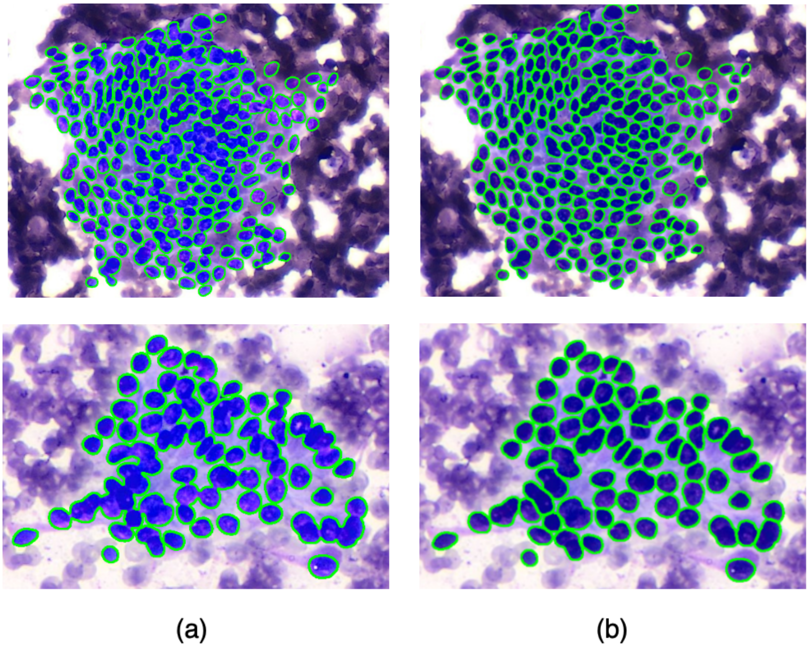

Figure 7: The illustration results were obtained by nnUnet (a) and the combined nnU-Net and Cellpose (b).

basic algorithms, which often struggle with time inefficiencies and inconsistent accuracy.

The outcomes of this research will be utilized to develop a classification framework, enabling physicians to diagnose the stage of thyroid cancer with greater accuracy and confidence. We also aim to collaborate with medical professionals to validate the method in real-world diagnostic scenarios, ensuring its practical utility. These advancements hold immense promise for revolutionizing automated cancer diagnostics, reducing the burden on healthcare providers, and ultimately improving patient outcomes through more precise and timely interventions in oncology.

References

[1] P. Bamford and B. Lovell, “Unsupervised cell nucleus segmentation with active contours”, *Signal processing*, 71(2), 1998, 203–13.

[2] J. Barker, A. Hoogi, A. Depeursinge, and D. L. Rubin, “Automated classification of brain tumor type in whole-slide digital pathology images

using local representative tiles", *Medical image analysis*, 30, 2016, 60–71.

[3] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm", *Computers & geosciences*, 10(2-3), 1984, 191–203.

[4] Y. Cheng, "Mean shift, mode seeking, and clustering", *IEEE transactions on pattern analysis and machine intelligence*, 17(8), 1995, 790–9.

[5] Y.-c. Cheng, W.-M. Qi, and W.-Y. Cai, "Dynamic properties of Elman and modified Elman neural network", in *Proceedings. International Conference on Machine Learning and Cybernetics*, Vol. 2, IEEE, 2002, 637–40.

[6] H. Fatakdawala, J. Xu, A. Basavanhally, G. Bhanot, S. Ganesan, M. Feldman, J. E. Tomaszewski, and A. Madabhushi, "Expectation–maximization-driven geodesic active contour with overlap resolution (emagacor): Application to lymphocyte segmentation on breast cancer histopathology", *IEEE Transactions on Biomedical Engineering*, 57(7), 2010, 1676–89.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn", in *Proceedings of the IEEE international conference on computer vision*, 2017, 2961–9.

[8] L. He, L. R. Long, S. Antani, and G. R. Thoma, "Histology image analysis for carcinoma detection and grading", *Computer methods and programs in biomedicine*, 107(3), 2012, 538–56.

[9] R. Hollandi, A. Szkalisity, T. Toth, E. Tasnadi, C. Molnar, B. Mathe, I. Grexa, J. Molnar, A. Balind, M. Gorbe, *et al.*, "nucleAIzer: a parameter-free deep learning framework for nucleus segmentation using image style transfer", *Cell Systems*, 10(5), 2020, 453–8.

[10] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation", *Nature methods*, 18(2), 2021, 203–11.

[11] J. W. Johnson, "Adapting mask-rcnn for automatic nucleus segmentation", *arXiv preprint arXiv:1805.00500*, 2018.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation", *IEEE transactions on pattern analysis and machine intelligence*, 24(7), 2002, 881–92.

[13] B. Korbar, A. M. Olofson, A. P. Miraflor, C. M. Nicka, M. A. Suriawinata, L. Torresani, A. A. Suriawinata, and S. Hassanpour, "Deep learning for classification of colorectal polyps on whole-slide images", *Journal of pathology informatics*, 8, 2017.

[14] Y. Miki, C. Muramatsu, T. Hayashi, X. Zhou, T. Hara, A. Katsumata, and H. Fujita, "Classification of teeth in cone-beam CT using deep convolutional neural network", *Computers in biology and medicine*, 80, 2017, 24–9.

[15] A. Mouelhi, M. Sayadi, F. Fnaiech, K. Mrad, and K. B. Romdhane, “Automatic image segmentation of nuclear stained breast tissue sections using color active contour model and an improved watershed method”, *Biomedical Signal Processing and Control*, 8(5), 2013, 421–36.

[16] N. Otsu *et al.*, “A threshold selection method from gray-level histograms”, *Automatica*, 11(285-296), 1975, 23–7.

[17] N. Otsu, “A threshold selection method from gray-level histograms”, *IEEE transactions on systems, man, and cybernetics*, 9(1), 1979, 62–6.

[18] H. A. Phoulady, D. B. Goldgof, L. O. Hall, and P. R. Mouton, “A new approach to detect and segment overlapping cells in multi-layer cervical cell volume images”, in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2016, 201–4.

[19] H. A. Phoulady, M. Zhou, D. B. Goldgof, L. O. Hall, and P. R. Mouton, “Automatic quantification and classification of cervical cancer via adaptive nucleus shape modeling”, in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, 2658–62.

[20] M. E. Plissiti, C. Nikou, and A. Charchanti, “Automated detection of cell nuclei in pap smear images using morphological reconstruction and clustering”, *IEEE Transactions on information technology in biomedicine*, 15(2), 2010, 233–41.

[21] H. Refai, L. Li, T. K. Teague, and R. Naukam, “Automatic count of hepatocytes in microscopic images”, in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, Vol. 2, IEEE, 2003, II–1101.

[22] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks”, *arXiv preprint arXiv:1506.01497*, 2015.

[23] J. B. Roerdink and A. Meijster, “The watershed transform: Definitions, algorithms and parallelization strategies”, *Fundamenta informaticae*, 41(1, 2), 2000, 187–228.

[24] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation”, in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, 234–41.

[25] R. Saha, M. Bajger, and G. Lee, “Spatial shape constrained fuzzy c-means (FCM) clustering for nucleus segmentation in pap smear images”, in *2016 international conference on digital image computing: techniques and applications (DICTA)*, IEEE, 2016, 1–8.

[26] O. Sarrafzadeh and A. M. Dehnavi, “Nucleus and cytoplasm segmentation in microscopic images using K-means clustering and region growing”, *Advanced biomedical research*, 4, 2015.

[27] H. Sharma, N. Zerbe, I. Klempert, O. Hellwich, and P. Hufnagl, "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology", *Computerized Medical Imaging and Graphics*, 61, 2017, 2–13.

[28] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation", *Nature methods*, 18(1), 2021, 100–6.

[29] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning", in *International conference on machine learning*, PMLR, 2013, 1139–47.

[30] A. Teramoto, H. Fujita, O. Yamamuro, and T. Tamaki, "Automated detection of pulmonary nodules in PET/CT images: Ensemble false-positive reduction using a convolutional neural network technique", *Medical physics*, 43(6Part1), 2016, 2821–7.

[31] J. P. Vink, M. Van Leeuwen, C. Van Deurzen, and G. de Haan, "Efficient nucleus detector in histopathology images", *Journal of microscopy*, 249(2), 2013, 124–35.

[32] L. Williams, "Pyramidal Parametrics", *ACM SIGGRAPH Computer Graphics*, 17(3), 1983, 1–11.

[33] K. Y. Win and S. Choomchuay, "Automated segmentation of cell nuclei in cytology pleural fluid images using OTSU thresholding", in *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, IEEE, 2017, 14–8.

[34] K. Wu, D. Gauthier, and M. D. Levine, "Live cell image segmentation", *IEEE Transactions on biomedical engineering*, 42(1), 1995, 1–12.

[35] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation", in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA/ML-CDS)*, Vol. 11045, *Lecture Notes in Computer Science*, Springer, 2018, 3–11.