Enhanced Preprocessing Techniques for Deep Learning in Pathology Image Analysis

Raghad ALMoqayad
2019008
Istanbul, Turkey
mahmoud.almoqayad@bahcesehir.edu.tr

Rayyan AL-HAJ
2017741
Istanbul, Turkey
ahmad.alhaj@bahcesehir.edu.tr

Mohammed Wael
2018383
istanbul,turkey
wael.adnan@bahcesehir.edu.tr

Abstract

This comprehensive report explores the development and implementation of sophisticated preprocessing techniques tailored for deep learning applications in pathology image analysis, with a special focus on whole slide images (WSIs). Central to this exploration is the innovation in stain normalization processes, where we adopted and refined the Macenko method, a critical step in ensuring visual consistency across diverse WSIs. We coupled this with advanced data augmentation strategies to address the inherent variability in medical images, thereby enhancing the robustness of subsequent deep learning models.

In tackling the challenge posed by high-resolution pathology slides, our methodology involved a meticulous approach to patch extraction. This was not merely about size reduction but ensuring that each extracted patch was representative of the crucial histopathological features. We further enhanced the image quality through nuanced intensity adjustments and selective thresholding, aiming to accentuate critical tissue characteristics while mitigating artifacts and inconsistencies.

A significant part of our work also entailed optimizing the preprocessing pipeline for scalability and efficiency. This was achieved through a methodical approach to batch processing and resource management, ensuring that our techniques could be applied to large datasets without compromising on processing speed or image quality.

The findings from this study underscore the pivotal role of precise and tailored preprocessing in medical image analysis. By elevating the quality and consistency of the input data for deep learning models, we significantly enhance their potential in accurate medical diagnosis, thus contributing to the broader field of digital pathology. This report not only details the technical aspects of our preprocessing methods but also contextualizes their impact in the realm of medical image analysis, paving the way for future advancements in the field.

1 Introduction

The advent of deep learning in medical image analysis heralds a transformative era in pathology, offering unprecedented tools for disease diagnosis and research. However, the effective utilization of deep learning models in pathology is intricately tied to the quality of image preprocessing. This report presents an in-depth exploration of advanced preprocessing techniques applied to whole slide

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

images (WSIs) in pathology, a domain marked by its rich yet complex visual data. The cornerstone of our approach is the advanced preprocessing of WSIs, crucial for accurate tissue analysis in medical diagnostics. Traditional methods often fall short in addressing the variability in tissue appearance due to staining procedures, which is a significant challenge in digital pathology. Drawing inspiration from existing literature, such as "A Comparative Study of U-Net Topologies for Background Removal in Histopathology Images," our project emphasizes the necessity of robust stain normalization. We adopt and enhance the Macenko method, a renowned technique for its effectiveness in normalizing histological stains, ensuring visual consistency across diverse samples. Our methodology extends beyond stain normalization. Recognizing the importance of high-quality data for training deep learning models, we have implemented sophisticated techniques for data augmentation and image quality enhancement. The project navigates through the challenges of processing high-resolution pathology slides, focusing on precise patch extraction. This step is not merely about size reduction; it involves ensuring that each patch encapsulates essential histopathological features. The extracted patches undergo rigorous quality checks, including removing patches with excessive whitespace and low tissue content, as highlighted in our custom scripts like ‘datacleaning.py‘ and ‘downsample.py‘. In addition, the report explores efficient processing techniques tailored for large datasets of medical images. We detail our approach to batch processing and resource management, crucial for scaling up the preprocessing pipeline without compromising on speed or image quality. This is particularly relevant given the growing size of medical image datasets in pathology. Our work is contextualized within the broader landscape of medical image analysis. We draw parallels and distinctions with existing studies, such as "Optimizing Storage and Computational Efficiency: An Efficient Algorithm for Whole Slide Image Size Reduction" and "Deep Learning-based Histopathological Segmentation for Whole Slide Images of Colorectal Cancer in a Compressed Domain," to underscore the unique contributions of our methodologies.

1.1 Literature Review

The literature pertinent to this project encompasses advancements in digital pathology, with a particular focus on image preprocessing and deep learning techniques.

1. "A Comparative Study of U-Net Topologies for Background Removal in Histopathology Images": This study explores various U-Net architectures for tissue segmentation in whole slide images (WSIs). It addresses the challenge of accurately identifying tissue regions amidst color variations and artifacts, highlighting the importance of background removal for precise analysis. The paper evaluates different U-Net topologies and their effectiveness in removing background from WSIs.

2. "Optimizing Storage and Computational Efficiency: An Efficient Algorithm for Whole Slide Image Size Reduction": This research focuses on an algorithm for efficiently managing WSIs by removing unneeded background and assembling tissue-containing parts into smaller images. The study highlights the algorithm’s ability to significantly reduce image size while maintaining diagnostic accuracy, demonstrating its potential in managing WSIs more efficiently.

3. "Preparing Data for Artificial Intelligence in Pathology": This paper discusses the importance of data preparation in the context of artificial intelligence applications in pathology. It underscores the necessity of quality data for effective AI analysis, emphasizing the role of preprocessing steps in enhancing data utility.

4. "Deep Learning-based Histopathological Segmentation for Whole Slide Images of Colorectal Cancer in a Compressed Domain": This study investigates a novel approach for pathology image segmentation in the compressed domain, aiming to reduce information loss and improve diagnostic accuracy. The paper presents a method that facilitates training neural networks on high-resolution images, demonstrating improved segmentation performance.

These studies collectively inform the project’s focus on effective image preprocessing techniques. They emphasize the crucial role of accurately preparing and processing medical images to enhance the performance of deep learning models, a fundamental aspect of modern digital pathology.

2 Methodology

The methodology of this project centers on the development of an advanced preprocessing pipeline for digital pathology images, with a focus on cleaning and structuring the data to enhance the performance of deep learning models.

2.1 Data preparation

The data preparation phase of the project, as delineated in the "downsampleext.py" and "dataprep.py" scripts, is pivotal in refining the quality of input data for deep learning analysis. This phase involves a series of intricate steps designed to extract and preprocess pathology image data effectively:

• Whole Slide Image (WSI) Handling and Patch Extraction: The dataprep.py script plays a critical role in extracting meaningful patches from WSIs. This involves reading WSIs and corresponding XML annotations to identify regions of interest (ROIs). The script calculates bounding boxes for these ROIs, ensuring the extraction of relevant tissue areas for analysis. It then proceeds to extract patches from these ROIs, adhering to specified dimensions and ensuring that each patch retains essential histopathological features.

• Downsampling and Data Augmentation: The downsampleext.py script focuses on downsizing the WSIs to manageable dimensions. This downsampling is not merely a reduction in image size but a careful preservation of critical details necessary for accurate medical analysis. This step is complemented by data augmentation techniques, which involve modifying the extracted patches to enhance the diversity of the dataset. This is crucial for training robust deep learning models that can generalize well on varied data. Both scripts incorporate stringent quality control measures. For instance, they include checks for excessive whitespace or low tissue content in patches, ensuring that only high-quality, information-rich patches are forwarded for further processing. The preprocessing steps also involve adjustments in image intensity and contrast, tailoring each patch to meet the standards required for effective deep learning analysis. By meticulously executing these steps, the data preparation phase sets a strong foundation for the subsequent stages of the project. It ensures that the deep learning models have access to high-quality, representative, and varied data, which is crucial for achieving accurate and reliable analysis in medical imaging.

2.1.1 Stain normalization

Stain normalization is a crucial preprocessing step in digital pathology, aiming to standardize the appearance of histological images that vary due to differing staining procedures. This standardization is vital for consistent analysis and comparison across multiple slides.

• Stain Normalization Process:

The "stainer.py" script is central to our stain normalization process. It begins by loading a reference image, which is crucial in setting a standard for color and intensity. This image is processed using the Macenko method, a well-regarded technique for its effectiveness in normalizing histological stains. The script then applies this normalization process to all the images in the dataset. Each image undergoes luminosity standardization before the stain normalization is applied, ensuring that the color representation in each image is consistent with the reference.

• Selection of Reference Image for Stain Normalization:

The "stain-select.py" script is designed to identify the most suitable reference image for stain normalization. This selection is critical as it sets the benchmark for the entire dataset's stain appearance. The code employs clustering algorithms and principal component analysis (PCA) to analyze the color features of a pool of images. By clustering these features, it identifies the most representative image, which is then used as the reference for normalization across all images. Both codes are implemented with attention to computational efficiency. "stainer.py", in particular, uses multiprocessing to expedite the normalization process across large datasets.The use of advanced image processing libraries ensures that the stain normalization is not only accurate but also retains the essential histopathological information necessary for subsequent deep learning analysis.

In summary, the stain normalization part of our methodology ensures that the variations caused by different staining protocols do not affect the deep learning model's ability to learn from and analyze

the histopathological images. This process is fundamental in creating a standardized, high-quality dataset that leads to more accurate and reliable results in medical image analysis.

2.2 Image Quality Enhancement

The image quality enhancement in our project, guided by "datacleaning.py" and "newsample.py", is a sophisticated process that employs a combination of technical strategies and computational tools to refine the dataset for deep learning analysis.

• Data Cleaning and Validation (datacleaning.py):

Utilizes OpenCV (cv2) for image processing tasks such as conversion to grayscale and thresholding. Implements custom algorithms to evaluate the color variance and intensity of image patches. Patches are analyzed to determine if their color variance exceeds a set COLORVARIANCETHRESHOLD and if their intensity meets the INTENSITYTHRESHOLD. Employs NumPy for numerical computations, especially in calculating the sum of binary threshold outputs to check against an AREATHRESHOLD, ensuring the patches have sufficient tissue content.

• Downsampling and Image Transformation (newsample.py):

Leverages OpenSlide for reading and handling WSIs and extracting specific regions based on XML annotations. Applies cv2 functions for bounding box calculations and image resizing to downscale the high-resolution images to a predefined TARGETSIZE, maintaining key histological details. Incorporates advanced resizing techniques, including aspect ratio preservation and padding, to retain the integrity of the image data post-downsampling.

• Image Quality Metrics:

Focuses on maintaining a balance between reducing image size for manageability and preserving sufficient detail for accurate pathological assessment. Adjusts thresholds and parameters based on empirical observations and dataset-specific requirements, ensuring the extracted patches are optimal for model training.

In summary, the image quality enhancement stage employs a combination of image processing libraries and custom algorithms to ensure that the dataset is not only technically sound but also optimized for extracting meaningful insights in subsequent deep learning analysis stages. This involves careful consideration of image characteristics, size, and quality, ensuring the prepared data aligns with the requirements of high-accuracy medical image analysis.

2.3 Efficiency and Scalability

Our approach to enhancing efficiency and scalability in image preprocessing, particularly in stain normalization, is encapsulated in the stainer.py and stainnew.py scripts. These scripts leverage advanced computational techniques and libraries to optimize the processing of large-scale histopathological datasets.

• Multiprocessing in Stain Normalization (stainer.py): Utilizes Python's multiprocessing library to parallelize the processing of images. This method involves creating a pool of worker processes, each handling a portion of the image normalization task, thereby significantly reducing overall processing time. The multiprocessing Pool is dynamically configured to use all available CPU cores, maximizing resource utilization.

• Advanced Stain Normalization (stainer.py): Employs the StainTools library for its core functionality, particularly using the Macenko method for stain normalization. This process involves fitting the normalizer to a reference image and then transforming other images in the dataset to match the reference's staining profile. LuminosityStandardizer from StainTools is used for pre-normalization image standardization, ensuring consistency in the luminosity of images before stain normalization.

• Intensity Adjustment and Staining Threshold Application (stainnew.py): Introduces additional image processing steps post-normalization. This includes adjusting image intensity using OpenCV's color space conversion and numpy for numerical operations. Applies a staining intensity threshold to further refine the image quality. This involves grayscale conversion and binary thresholding to isolate areas of significant staining.

• Efficient Batch Processing: Both scripts are designed for efficient batch processing of images. This is particularly important when dealing with large histopathological datasets, where individual image processing can be time-consuming. In summary, the stainer.py and stainnew.py scripts represent a technically sophisticated approach to stain normalization, combining advanced image processing techniques with multiprocessing for improved efficiency and scalability. This methodology ensures that large datasets of histopathological images are processed quickly and consistently, preparing them for accurate and reliable analysis in deep learning models.

2.4 Pipeline shredding

Our methodology for 'pipeline shredding' is exemplified in the "stainnew.py" script, which epitomizes the breaking down of the image preprocessing workflow into distinct, methodical steps. This process ensures the highest quality of processed images for deep learning applications in pathology.

• Reference Image Standardization: The script begins by loading a pre-selected reference image for stain normalization. This image undergoes luminosity standardization using StainTools' Luminosity-Standardizer, ensuring that the lighting conditions of the reference image match those of the images to be processed and Stain Normalization utilizes StainTools' StainNormalizer with the Macenko method. The normalizer is fitted to the standardized reference image, setting a benchmark for the staining appearance in the dataset.

• Intensity Adjustment: A key feature in the script is the adjustintensity function. It adjusts the intensity of each image post-normalization, using OpenCV for color space manipulation. The function converts images to the HSV color space and adjusts the V (Value) channel by a specified factor, enhancing or reducing image brightness as required.

• Staining Threshold Application: The applystainingthreshold function applies a binary threshold to the image. It converts the image to grayscale and then uses OpenCV's thresholding function to isolate areas with staining intensity above a specified threshold. This step is crucial for emphasizing significant staining features while discarding less relevant areas.

• Multiprocessing for Efficient Processing: The script employs Python's multiprocessing to handle large datasets efficiently. By distributing the image processing tasks across multiple CPU cores, the script maximizes resource utilization and minimizes processing time.

• End-to-End Process Management: The script is designed to manage the entire process from loading and processing each image in the dataset to saving the normalized output. This includes error handling to ensure robustness in processing large datasets.

This shredding of the pipeline into specific, well-defined tasks allows for greater control over each step of the image preprocessing process. It ensures that each image is treated individually with the necessary adjustments and checks, leading to a dataset that is homogenous in terms of stain appearance and optimized for subsequent analysis

3 Results

Upon completion, this project was anticipated to yield the following results:

3.1 Quantitative Metrics in Image Quality

A detailed analysis of pre- and post-processing images would likely reveal quantitative improvements in key image quality metrics such as contrast, sharpness, and color fidelity. These metrics are crucial for ensuring that deep learning models accurately interpret tissue morphology.

3.2 Data Throughput

With optimized algorithms, we would expect to see a substantial increase in data throughput, quantified by metrics like images processed per second, demonstrating the efficiency of our multiprocessing approach.

3.3 Stain Normalization Consistency

The effectiveness of the stain normalization process would be measured by comparing the color histograms of processed images against the reference image, aiming for minimal deviation.

4 Discussion

4.1 Reference Studies Impact

Drawing insights from "Efficient Algorithm for Whole Slide Image Size Reduction" and other reference papers, the project's methodologies would contribute to the field's understanding of efficient WSI processing. The balance between size reduction and detail preservation would be a key area of analysis.

4.2 Challenges and Solutions

The project would likely encounter specific challenges in maintaining color consistency across various stains and tissue types. Addressing these challenges might involve iterative tuning of normalization parameters or exploring alternative normalization techniques.

4.3 Implications for Deep Learning Models

The project's preprocessing steps, particularly in data cleaning and normalization, would have significant implications for the accuracy of deep learning models in pathology. The project would serve as a case study for the impact of data preprocessing on model performance, especially in fields requiring high-precision image analysis like medical diagnostics.

5 Conclusion

This project marks a notable step forward in digital pathology, emphasizing image preprocessing for deep learning. It introduces methods in Python scripts for stain normalization, image quality, and processing large datasets. Time limits and lack of expert input restricted exploring these methods fully.

The project underscores the balance between technical precision and practicality in medical image analysis. It demonstrates the importance of preprocessing in improving AI accuracy in diagnostics. Future work aims to automate parameter adjustments and enhance real-time analysis, advancing the field further.

References

1- Kim, H., Yoon, H., Thakur, N., Hwang, G., Lee, E. J., Kim, C., Chong, Y. (2021). Deep learning-based histopathological segmentation for whole slide images of colorectal cancer in a compressed domain.

2- Faghani, S., Codipilly, D.C., Moassefi, M., Iyer, P.G., Erickson, B.J. (2023). Optimizing Storage and Computational Efficiency: An Efficient Algorithm for Whole Slide Image Size Reduction.

3- Riasatian, A., Rasoolijaberi, M., Babaei, M., Tizhoosh, H.R. (2020). A Comparative Study of U-Net Topologies for Background Removal in Histopathology Images.

4- Alexander Selvikvåg Lundervold, Arvid Lundervold. (2019). An overview of deep learning in medical imaging focusing on MRI.

5- Mustafa Umit Oner, Mei Ying Ng,Danilo Medina Giro. (2022). An AI-assisted tool for efficient prostate cancerdiagnosis in low-grade and low-volume cases.

6- Liron Pantanowitz, Gabriela M Quiroga-Garza, Lilach Bien, Ronen Heled, Daphna Laifenfeld, Chaim Linhart, Judith Sandbank. (2020) . An artificial intelligence algorithm for prostate cancer diagnosis in whole slide images of core needle biopsies: a blinded clinical validation and deployment study