Project Report: Automatic Polyp Segmentation in Colonoscopy Images Using U-Net Architecture

Abstract

Colorectal cancer represents a significant global health challenge, with early detection and removal of polyps during colonoscopy procedures serving as the most effective method for prevention. This project focuses on the development and implementation of a deep learning pipeline designed to automatically segment polyps from colonoscopy images. Utilizing a U-Net architecture implemented in Python with the PyTorch framework, the system processes biomedical imagery to produce binary segmentation masks. The project emphasizes a modular, reproducible workflow, incorporating automated data acquisition, rigorous preprocessing, and a custom neural network architecture. Despite computational constraints necessitating the use of central processing unit (CPU) training, the model demonstrated rapid convergence and successfully learned to identify polyp regions, achieving a Dice Coefficient of approximately 0.57 after limited training epochs.

1. Introduction

The gold standard for colorectal cancer prevention is the colonoscopy, a procedure allowing gastroenterologists to visually inspect the bowel for anomalies known as polyps. While effective, the procedure is highly dependent on the operator's skill. Factors such as lighting conditions, the variety of polyp shapes, and the presence of debris can lead to missed detection rates. To mitigate this, Computer-Aided Diagnosis systems have emerged as a supportive tool to draw the clinician's attention to potential lesions.

The primary objective of this project was to engineer a semantic segmentation pipeline capable of delineating polyps from the surrounding mucosa at the pixel level. Unlike simple classification, which merely indicates the presence of a polyp, segmentation provides the precise location and boundary, which is clinically more valuable for surgical planning and size estimation. The system was built using modern deep learning techniques, specifically the U-Net convolutional neural network, which is widely regarded as the state-of-the-art architecture for biomedical image segmentation.

2. Data Acquisition and Curation

The foundation of any machine learning model is the quality of its training data. This project utilized the Kvasir-SEG dataset, an open-access dataset provided by Simula Research Laboratory, which contains 1,000 distinct images of polyps along with their corresponding ground-truth segmentation masks annotated by medical experts.

To ensure the reproducibility of the scientific pipeline, a custom automation script was developed in Python. Rather than relying on manual downloads, which are prone to user error and file corruption, the script programmatically fetches the dataset from the source. A significant technical challenge encountered during this phase was the strict Secure Sockets Layer (SSL) verification on the host server, which frequently blocked automated requests. This was resolved by implementing a robust request handler that bypasses SSL certificate verification, ensuring consistent access to the data regardless of the local network security configuration.

3. Methodology: Data Preprocessing

Raw medical images are rarely suitable for direct ingestion into a neural network. A systematic preprocessing pipeline was implemented to standardize the input data. The first step involved spatial standardization. While medical images vary in resolution, deep learning models require uniform input dimensions. All images and masks were resized to 128 by 128 pixels. This resolution was selected as a strategic trade-off; while higher resolutions such as 256 or 512 pixels preserve finer details, they exponentially increase the memory required for processing. Given that the training environment was restricted to a CPU rather than a Graphics Processing Unit (GPU), reducing the resolution allowed for efficient training iterations without exhausting system memory.

Following resizing, the images underwent normalization. The pixel intensity values, originally ranging from 0 to 255 in the RGB color space, were scaled to a floating-point range between 0 and 1. This step is critical for numerical stability, as it prevents exploding gradients during the backpropagation process. Simultaneously, the ground-truth masks were binarized. Since the task is a binary classification problem—distinguishing pixels as either polyp or background—the masks were thresholded to ensure they contained only values of 0 or 1, eliminating any interpolation artifacts introduced during the resizing process. Finally, the data was permuted into the NCHW format (Batch, Channels, Height, Width) to align with the tensor requirements of the PyTorch framework.

4. Methodology: Model Architecture

The core of the system is the U-Net architecture, a fully convolutional network designed specifically for biomedical tasks where labeled data is often scarce. The architecture derives its name from its U-shaped structure, which consists of two symmetric paths: the encoder and the decoder.

The encoder, or contracting path, functions as a feature extractor. It is composed of a series of convolutional blocks followed by max-pooling operations. As the image progresses through the encoder, its spatial dimensions are reduced by half at each step, while the number of feature channels doubles. This allows the network to capture high-level semantic context, effectively answering the question of what is present in the image, albeit at the cost of spatial precision.

The decoder, or expansive path, is responsible for precise localization. It utilizes transposed convolutions to upsample the feature maps back to the original image resolution. The critical innovation of U-Net is the use of skip connections. These connections transfer high-resolution feature maps directly from the encoder to the corresponding layers in the decoder. By concatenating these features, the network combines the semantic context from the decoder with the fine-grained spatial information from the encoder, allowing for the precise delineation of polyp boundaries.

The final layer of the network utilizes a 1x1 convolution with a Sigmoid activation function. This transforms the network's output into a probability map, where every pixel is assigned a score between 0 and 1 representing the likelihood of that pixel belonging to a polyp.

5. Implementation Strategy and Training

The project was initially explored using the R programming language; however, it was migrated to a native Python environment utilizing the PyTorch library. This transition was driven by the need for a more robust and industry-standard deep learning ecosystem. Python offered superior compatibility with system-level C++ libraries and eliminated the dependency conflicts often found when bridging R with deep learning backends on Windows operating systems.

The training process was orchestrated using the Binary Cross-Entropy (BCE) loss function. BCE is the mathematical standard for binary classification tasks, effectively penalizing the model when its predicted probabilities diverge from the ground truth. The optimization of the network weights was performed using the Adam optimizer, an algorithm that adapts the learning rate for each parameter. This allowed the model to converge more quickly than standard Stochastic Gradient Descent.

The model was trained for three epochs with a batch size of 16. A random split was applied to the dataset, reserving 800 images for training and 200 images for validation. This separation ensured that the model's performance was evaluated on unseen data, providing a realistic measure of its generalization capabilities.

6. Results and Evaluation

The performance of the system was evaluated quantitatively using the Dice Coefficient, also known as the F1-Score. In semantic segmentation, standard pixel accuracy can be misleading because the background often occupies the majority of the image. The Dice Coefficient provides a more rigorous metric by calculating the area of overlap between the predicted segmentation and the ground truth, divided by the total number of pixels in both.

Upon completion of the training phase, the model achieved a training loss of approximately 0.39 and an average Dice Coefficient of 0.57 on the validation set. These results indicate that even with a reduced resolution and limited training epochs, the network successfully learned to generalize the features of polyps. Visual inspection of the results confirmed that the model could correctly locate the primary polyp structures, although the boundaries lacked the smoothness that would likely be achieved with higher-resolution inputs and extended training time.

7. Conclusion

This project successfully demonstrated the implementation of an end-to-end deep learning pipeline for medical image analysis. By leveraging the U-Net architecture and a modular software design, the system proved capable of segmenting colorectal polyps with promising accuracy. The work highlighted the importance of robust data preprocessing and the effectiveness of modern convolutional neural networks in extracting semantic features from complex biomedical imagery. Future improvements to the system would involve deploying the pipeline on GPU-accelerated hardware to support higher image resolutions and implementing data augmentation techniques such as rotation and elastic deformation to further enhance the model's robustness against clinical variations.