Gastrointestinal Polyp Segmentation and Classification

Muhammad Kabir Hamzah, Beyza Zayim, Aya Elgebaly, Faisal Farhan, Diptan Regmi

Abstract

The rate of colorectal cancer is growing at an alarming rate all over the world. The survival rate of patients drastically decreases when the cancer reaches its final stage. Hence the identification and removal of colon polyps is very important for the patients at early stages. Advanced computer-aided systems can assist polyp detection that outperforms manual detection methods. In this project we present the development of an automated polyp detection system using advanced image processing algorithms, machine learning and deep learning models. The first pipeline showcases the automated segmentation of polyps using advanced image processing methods. It achieved an overall detection accuracy of 79.47% with an average dice coefficient of 46.67%. The second group of pipelines involved classification of polyps using machine learning models and deep learning models. We explored two methods of image feature extraction for the machine learning classifiers namely, Local Binary pattern (LBP) and Gray Level Co-occurrence matrix (GLCM). The LBP based feature extraction, with LightGBM classifier gave the best accuracy for polyp classification, achieving 94% accuracy and 93.9% F1-score. In comparison to the machine learning classifier, the deep learning classifier achieved upto 96% test accuracy. The last pipeline employed deep learning models for polyp segmentation, with the best model achieving up to 90.76% and 72.13% dice coefficients on the validation and test datasets respectively. Overall, we confirmed that the deep learning pipeline gave promising results for both polyp segmentation and classification.

Keywords: Polyp detection; Polyp segmentation; Advanced Image Analysis; Machine Learning; Deep Learning

1. Introduction

Colorectal cancer (CRC) which is a type of cancer that originates from the colon or rectum has become a worldwide health concern over the years as it is now termed as the second leading cancer related death globally. Recent research has shown that almost 95% of colorectal cancer originates from polyps which is an abnormal tissue growth in the gastrointestinal (GI) tract [1,2]. The colorectal polyps may be unnoticeable in the initial stage, but the survival stage of patients drops to less than 5% when they reach the final stage. As the growth of polyps in the GI tract is a risk factor for cancer, it is crucial to detect and remove polyps to prevent the development of colorectal cancer. Although there are advanced medical diagnosis systems like colonoscopy, endoscopy for detecting polyps, these methods have some drawbacks as well. Previous studies have shown that in many cases the procedure requires expensive medical resources, and the manual procedure creates discomfort to the patients. Moreover, the diagnosis relies completely upon the doctor's experience and the detection of polys may sometimes go unnoticed due to the variation in size, color, structure of the polyps [1]. Manually detecting the polyps also requires a lot of time and effort. Hence, introducing an automated and robust technique for detecting polyps in the GI tract can aid pathologists to detect early stages of colorectal cancer more accurately and efficiently. With the advancement of computer aided diagnosis methods and development in the medical field, promising results have been seen for automatic diagnosis of human diseases in different parts of the human body including gastrointestinal stomach [3]. Several research have been conducted over the years using some

current state of the art methods ranging from traditional machine learning algorithms to recent advanced deep learning methods for automatic detection of abnormalities in the gastrointestinal tracts [4]. Advanced image segmentation and detection techniques can assist pathologists in many ways for detecting abnormalities and improving the diagnostic ability for early detection of polyps and preventing colorectal cancer. In this project, we applied advanced image processing methods as well as traditional machine learning and deep learning techniques to preliminary identify the presence of polyps and segment them.

2. Literature Review

Over the years, several research works have been conducted using computer aided systems for colorectal polyp segmentation. In the paper [5] Canny edge detector was applied for processing the pre-filtered polyp dataset and the relevant edges were identified using a template matching technique for polyp segmentation. Authors in [6] used a new type of framework implementing two different types of SegNet architecture for polyp segmentation. A real time polyp identification and delineation was proposed in [7]. The proposed method was named ColonSegNet that can minimize miss detection rates of polyps efficiently. Authors in [8] proposed two sequentially stacked pretrained encoder-decoder networks for polyp segmentation, with the second network receiving the first network's output as input. The first network's initial prediction serves as an attention mechanism, allowing the second network to focus on interesting areas within the image, improving the accuracy of its predictions. A dual branch multiscale feature fusion network was proposed in [9] for polyp segmentation that implements CNN and transformer networks in parallel for extracting multiscale local information and global contextual information. A new architecture was developed by Sanderson et al. [10] that merges FCNs and transformers for polyp segmentation in colonoscopy images. In [11], the authors proposed a parallel reverse attention network for polyp segmentation, with deep supervision for the last three side outputs. Shen et al. [12] proposed a new hard region enhancement network for polyp segmentation with supervised multi-level ground truth. In the paper [13], the authors suggested a temporal correlation network for video polyp segmentation, in which the temporal correlation is unprecedentedly modelled based on the relationship between the original video and the captured frames to be adaptable for video polyp segmentation. Chen et al. [14] proposed a novel two-branched network for polyp segmentation from single-modality endoscopic images, consisting of an image-to-image translation branch and an image segmentation branch.

3. Dataset Description

In this experiment, a few different publicly available datasets were used. These datasets consist of polyp images that consist of ground truth and corresponding segmentation masks. They are described in Table 1.

Dataset	Task Type	Quantity
CVC-ColonDB	SEGMENTATION	380
CVC-ClinicDB	SEGMENTATION	62
Etis-LaribPolypDB	SEGMENTATION	196
Gastro-vision	CLASSIFICATION	1789

Table 1: Dataset Description

CVC-ColonDB: The CVC-ColonDB[15] is a colonoscopy image database specifically curated for the development and evaluation of automated polyp detection and segmentation algorithms. It comprises of 380 (574×500) images describing different types of polyps with annotated polyp boundaries, meticulously marked by expert endoscopists.

CVC-ClinicDB: The CVC-ClinicDB [16] database, also a collection of colonoscopy image dataset curated for the development and benchmarking of automated polyp detection and segmentation algorithms. It contains 62 (384×288) images with annotated masks indicating polyp areas.

Etis-LaribPolypDB: The dataset [17] contains 196 (1255x966) images and their corresponding masks.

Gastro-vision: Gastro-vision [18] dataset is a comprehensive multi-class endoscopy image collection, featuring 8,000 images from the upper and lower gastrointestinal (GI) tracts. The colon polyp and normal stomach classes account for 1789 of the 8000 images in the database.

For machine learning and deep learning-based classification, we use Gastro-vision for training, validation, and testing.

For the deep learning-based segmentation we combine CVC-ColonDB and Etis-LaribPolypDB as the training and validation set while using CVC-ClinicDB as the test set.

For the image processing-based segmentation, we opted to use CVC-ColonDB as it contains images with the most visual descriptive features, although we had to create a subset as discussed in the image processing part.

4. Methodology

As stated earlier, our work contains different pipelines for both classification and segmentation of colon polyps. The first pipelines involve the use of advanced image analysis for the end-to-end segmentation of colon polyps. The second pipeline implements two different feature extraction algorithms followed by machine learning classifiers to classify normal stomach and colon polyp. The third pipeline uses features extracted by image analysis to classify colon polyps and then implementing deep learning-based classification and segmentation of colon polyps respectively.

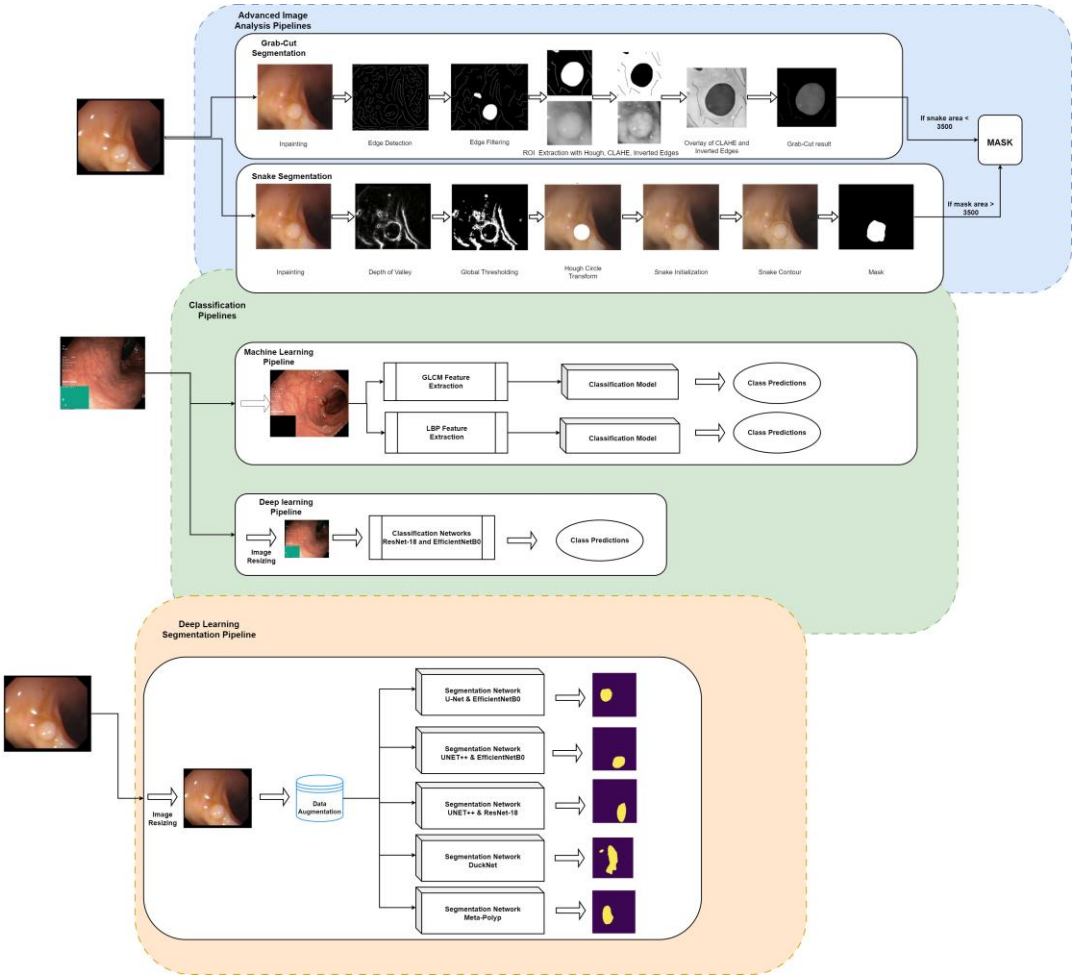

Figure 1. Workflow diagram of the different pipelines

4.1 Advanced Image Analysis

4.1.1 Dataset Description

The database contains 380 images with corresponding masks of polyps' regions which we believe to have been obtained from 17 patients. This conclusion is justified by the difference in the appearance of the polyps across the entire data. We clustered these images based on the patients to obtain the representative samples shown in figure 2.

Although each cluster of images were obtained from the same patient, they do have some distinct differences which are mainly due to:

1. Orientation of the endoscope during acquisition
2. Distance of the camera to the object
3. Illumination during acquisition

The descriptive features of polyps for each of these patients are significantly different. Two of which pose the most challenge are the shapes and sizes, other distinguishable features include the contrast, color, and texture.

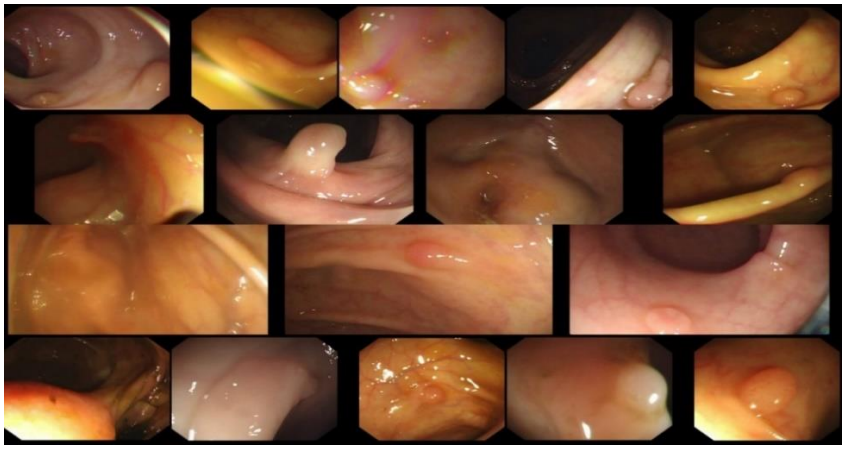

Figure 2. Representative samples from CVC-ColonDB Database

From the total of 380 images, we remove all extreme cases of polyps leaving about 190 images, these extreme cases were:

1. Those extremely large that they cover the entire image.
2. Those extremely small that they aren't visible.
3. Blurry image.

Images where polyps are located at the boundaries of the image.

4.1.2 Image Analysis for Polyp Segmentation

In the following sections, we outline the preprocessing steps and algorithms developed for the automatic segmentation polyps using advanced image analysis methods on the Colon-DB data. Figure 3. provides a schematic representation of the various stages involved in the Grab-cut segmentation algorithm (pipeline), while figure 8 illustrates the workflow of the active contour segmentation (pipeline two). Finally, the third section explains how these two methods were combined to achieve the final segmentations (pipeline three).

4.1.2.1 Polyp Segmentation with Grab-cut

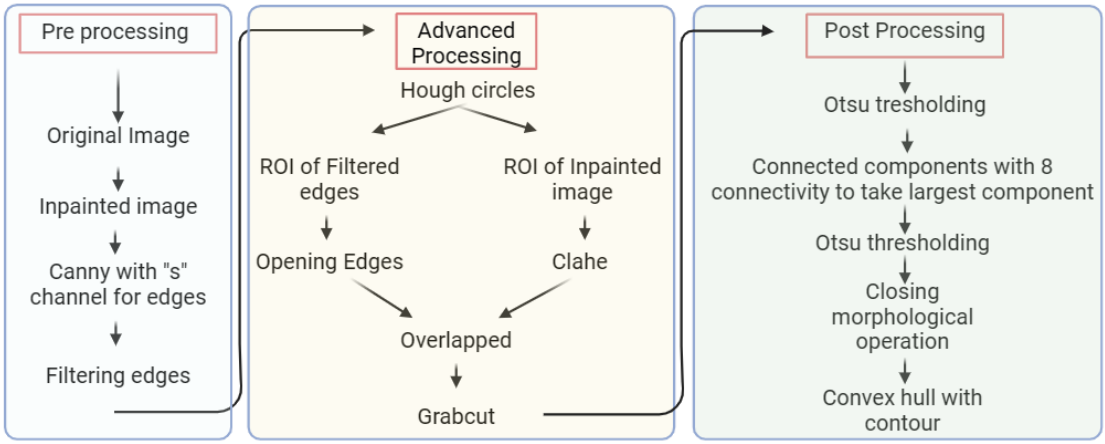

Figure 3. Workflow of the Grab-cut technique

I) Pre-Processing

In this system, the preprocessing stage is designed to enhance the quality of the endoscopic images, making segmentation easier. This step eliminates black frames and removes specular highlights.

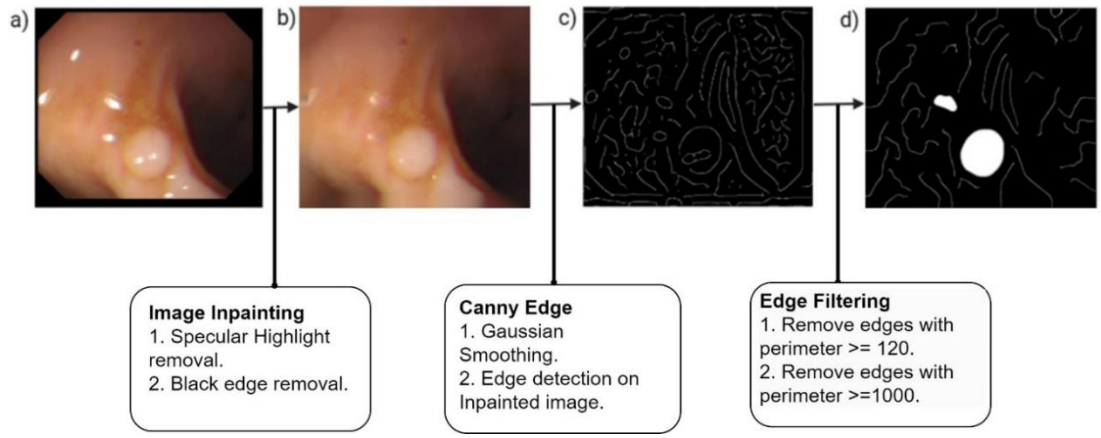

Figure 4. a) Original image, b) Inpainted image, c) Canny edges result, d) Filtered edges result

a. Image Inpainting

Specular highlight removal:

The first step of the pre-processing phase was to identify and remove specular highlights/reflections. Specular highlights are generally bright spots of light that appear on shiny or reflective surfaces when they reflect a light source. These highlights occur because the surface reflects the light in a mirror-like manner, causing the light to appear concentrated and much brighter than the surrounding areas.

In our case, specular highlight removal enhances the image resulting in a cleaner and more consistent visual representation. The process was achieved by:

- Identifying Specular Highlights: The images were converted from BGR (Blue, Green, Red) color space to HSV (Hue, Saturation, Value) color space. HSV is often used in image processing because it separates the intensity (Value) from the color information (Hue and Saturation). The Saturation channel was threshold-ed to create a binary image with a threshold value of 20%. The Value channel was also threshold-ed to create another binary image with a threshold value of 50%. We merged the saturation and value channels to obtain the specular highlight regions.
- Dilation: A dilation operation using a 3x3 kernel was performed on the specular highlight regions to increase the area of small and scattered highlights. image using a 3x3 kernel. This process ensures that the entire specular regions have been captured.
- Inpainting to Fill Highlighted Regions: Finally, we inpainted the original image using the mask obtained above. The Telea [19] inpainting algorithm was used for this purpose. Inpainting fills in the highlighted regions to eliminate the specular highlights.

Black edge removal:

Black edges are regions around the boundaries of the image that appear as black pixels. It was necessary to be dealt with as it introduces artifacts during canny edge detection. The process was accomplished by:

- Identifying black edges: We use value (v) channel to create a binary mask for black regions using a threshold of 7%. This binary mask was inverted so that the dark regions from the original image appear as white, indicating the regions to be used by the inpainting algorithm.
- Dilation: Just as in the specular highlight removal, the inverted mask undergoes a dilation operation to increase the area of the masked regions.
- Inpainting to Fill Dark Regions: We again applied the Telea inpainting algorithm to obtain the final inpainted image as shown in figure 4 (a).

b. Canny edge detection

We apply the canny edge detection algorithm on the inpainted image using a gaussian filter with standard deviation (σ) value set to 5. The sigma value controls the amount of smoothing or blurring applied to the image before edge detection. The lower and higher threshold range was set to (0.25, 0.65).

The Canny detection algorithm was applied particularly in the saturation channel (s) of the inpainted image. Saturation, in the context of color, refers to the intensity or purity of a color, which represents the amount of gray in proportion to the hue. Highly saturated colors are vivid and intense. So, it was used for edge detection as it enhances the visibility of edges in an image.

c. Filtering of edges

The result of the canny edge detection was filtered out by considering the perimeter of all objects detected. Objects whose perimeter weren't in the range 120-1000 (typical polyp perimeter) were not considered.

II) Grab-Cut Segmentation Algorithm

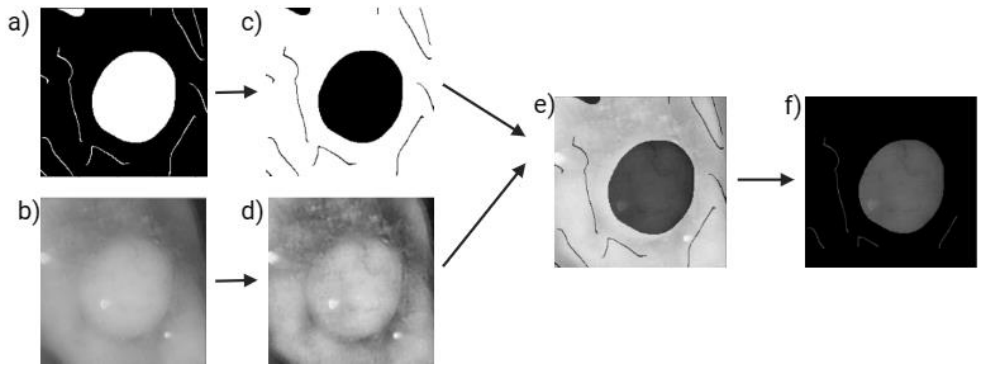

Figure 5. Grab-cut Segmentation Steps. a) ROI of filtered edges, b) ROI of inpainted image, c) Morphological Opening operation and inversing, d) Result of CLAHE, e) Overlap of CLAHE and Inverted edges, and e) Grab-cut result

a. Hough Circles

Hough Circle transform is a useful technique that localizes circular structures in an image. Since polyps are typically elliptical in shape, we use the Hough Circles to detect objects in the filtered edges with elliptical shapes. The center and radius of these circles were used as coordinates to extract squares of size 1.6 times the radius. Enlarging the radius was necessary to capture the entire polyp area. These squares were used as regions of interest (ROIs) in subsequent steps.

i. ROI of Filtered edges and opening edges
   The filtered edges result (Fig. 4(d)) was used as an input of the Hough Circles detection to obtain the ROI as indicated by figure 5a. We inversed and applied an opening operation on the ROIs of the filtered edges to close some of the detected edges as shown in fig 5c.
ii. ROI of Inpainted image and Histogram Equalization
   The inpainted image result (Fig. 4(b)) was used as an input to the Hough Circles detection to obtain the ROI as indicated by figure 5(b). To improve the contrast of the images (which is necessary for the Grab-cut performance), we applied the Contrast Limited Adaptive Histogram Equalization (CLAHE) on the ROIs of the inpainted images.

b. Grab Cut

Grab-Cut is an image segmentation method that iteratively refines a segmentation of an image into foreground and background regions. We overlapped the result of CLAHE with the edges as obtained in fig 5c. The result of these overlaps as shown in fig 5.e was used as input to the grab cut algorithm. The following indicates the steps involved in the grab cut algorithm.

i. **Foreground and background pixels:** These models store information about the color distribution of the background and foreground pixels.
ii. **Rectangle of the Grab cut:** A rectangle defining the initial estimate of the foreground region. This rectangle should loosely enclose the object of interest. In our case we specified the rectangle as the same size as the ROI patches extracted.
iii. **Application:** 10 iterations were applied to refine the segmentation. For each iteration, the Grab-Cut algorithm updates the mask to label each pixel as; 0 (Background), 1 (Foreground), 2 (Probably background), 3 (Probably foreground). The result is an image (Fig 5 f) where only the polyp is visible, and the background is set to black.

III) Post-processing

a. Otsu Thresholding

To convert the grab cut results to binary masks, we applied Otsu thresholding to achieve this and to prepare the image for the next processing.

b. Connected Components

In some cases (such as the examples illustrated by figure 6), we observed artifacts in the threshold-ed masks. We use connected component analysis on these masks with 8-way connectivity, to return only the largest component in the image as polyp masks.

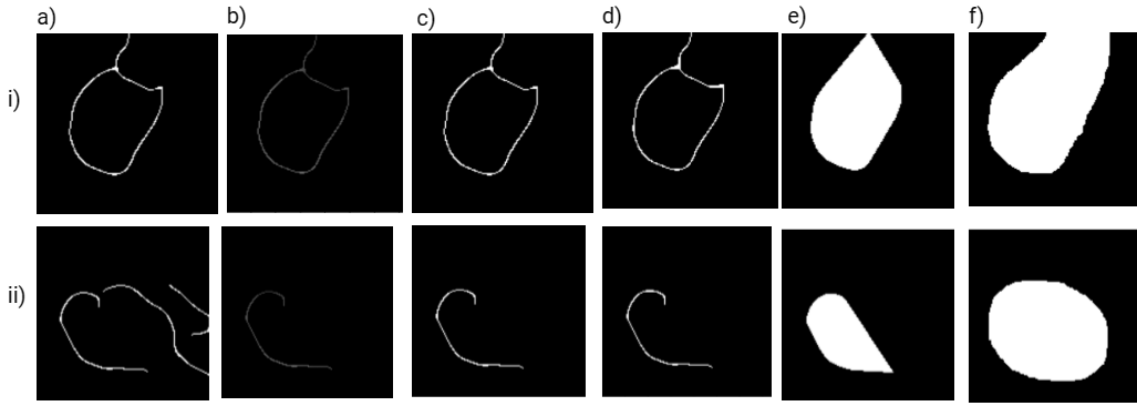

Figure 6. Post Processing steps to obtain the final segmentation. a) Otsu thresholding results, b) Largest Connected Components, c) Otsu thresholding, d) Morphological Closing operation, e) Convex hull results, f) Ground truths

c. Morphological Closing Operation

In other cases, we observed that the grab cut algorithm failed to identify the masks, resulting in only partial edges. We applied the morphological closing operation to deal with such cases.

d. Convex hull

Where the morphological closing fails, a convex hull was applied. This approximates the masks using the smallest possible convex polygon (or polyhedron) to enclose all the points. To obtain the final segmentation, we used the coordinates of the earlier extracted ROI patches to revert back to the original shape. This procedure allows for computing the dice coefficient with respect to the ground truth as well as visualization of the exact locations of the polyps.

The grab-cut achieved a detection accuracy of 54.74% and an average segmentation dice score of 29.04 %.

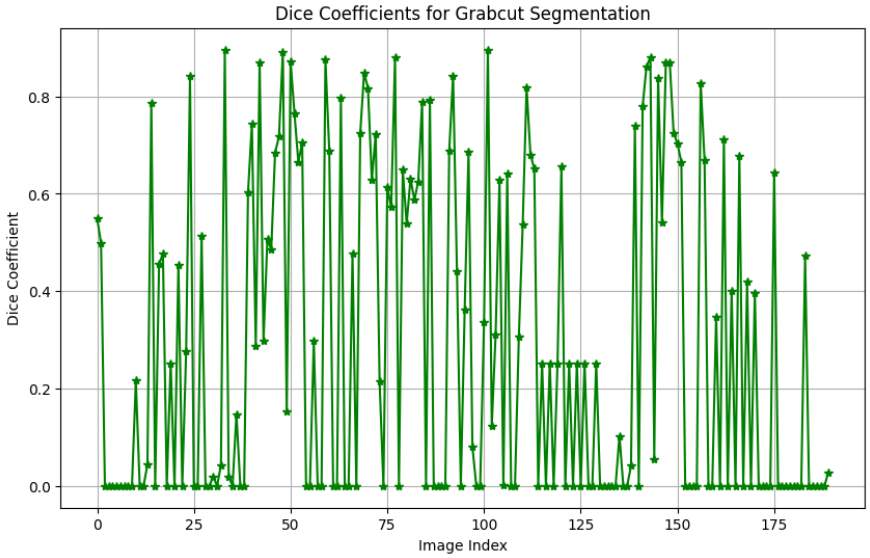

Fig 7. Dice Coefficients of the Grab-Cut Segmentation across 190 images

4.1.2.2 Polyp Segmentation with Active Contour

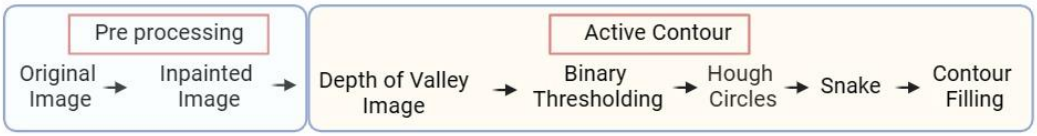

Figure 8. schema of the active contour algorithm.

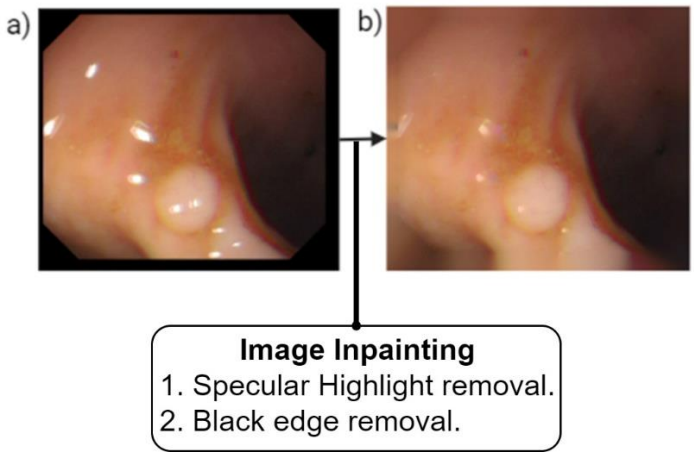

Figure 9. Image pre-processing for snake segmentation. a) Original Image, b) Inpainted image

Active contours, also known as snakes, are a method used in image processing to detect and outline object boundaries within an image [20]. They are particularly useful for tasks such as segmentation, where the goal is to delineate regions of interest.

Like the preprocessing phase of the Grab-cut technique, the preprocessing phase of the active contour involves the image inpainting (specular highlight and black edge removal). Next, we discuss the steps (as shown in fig 8) involved in segmenting polyps using active contours.

Algorithm Design

a. Depth of valley image

The Depth of Valley (DoV) image focuses on identifying and enhancing the valleys, or regions of low intensity, between areas of higher intensity in an image. These valleys often correspond to boundaries between different anatomical structures or regions of interest. By emphasizing these valleys, the DoV method improves the accuracy and robustness of segmentation algorithms.

It is useful for enhancing the detection of boundaries and regions of interest, particularly in images where standard edge detection methods might struggle. This approach is especially useful in our case where it helps to properly outline structures of polyps. To obtain the DoV image, we adhered to the methodology of [21] which involves:

i. **Image Smoothing:** We applied a smoothing filter (Gaussian blur) on the inpainted image, to reduce noise and minor variations in intensity that could interfere with valley detection.

ii. **Gradient Computation:** We used the Sobel filter to compute the gradient of the image to identify regions with significant changes in intensity. This step highlights potential boundaries and regions of the polyps.

iii. **Valley Detection:** We use the Sato filter on the Gaussian smoothed image to identify ridges in the image which corresponds to edges of objects with rapid intensity changes.

iv. **Image Smoothing:** We applied a smoothing filter (Gaussian blur) on the inpainted image, to reduce noise and minor variations in intensity that could interfere with valley detection.

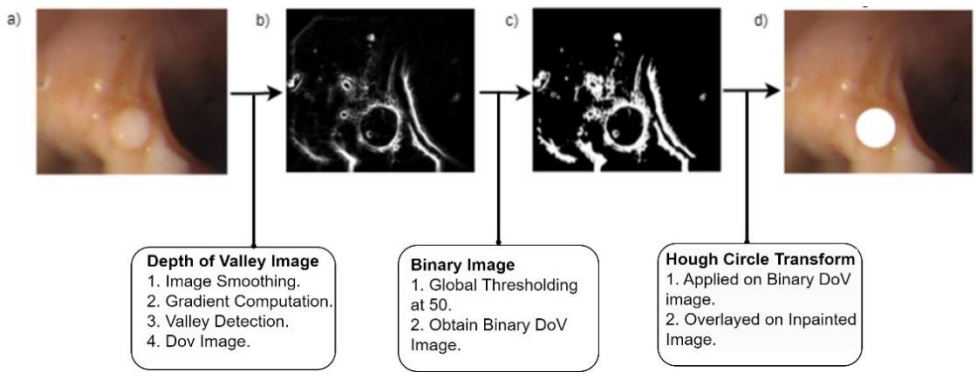

Figure 10. Polyp ROI acquisition. a) Inpainted Image, b) DoV image, c) Binary DoV image, d) Hough Circle result

b. Binary Thresholding

To convert the DoV images to binary images, we applied a binary thresholding with a threshold set to 50 to suppress regions that aren't polyps but highlighted in the DoV image. The result of this can be seen in Fig 10(c).

c. Hough Circles

Like the Hough Circles algorithm in the Grab-Cut segmentation, we applied Hough Circles on the binary DoV images since boundaries of polyps in the DoV image also appear elliptical in shapes. The coordinates of the circles are later used as seed points for the snake segmentation algorithm. The Hough circle output can be seen in Fig 10(d).

d. Snakes

Using the coordinates of the circles as seed points (initialization) and the binary DoV image, we parameterized the active contour algorithm by considering how much importance we give to edges (attraction to edges), snake length, snake smoothness, and attraction to brightness. We tuned each of these parameters to allow the maximum possible segmentation across all images. Since our input image can typically be considered as an edge image, we weighted the "attraction to edges" parameter more than the others. An example of the result of the snake segmentation is as shown in figure 11(b).

We filled the snake contours to obtain the final segmentation and to efficiently compute the dice coefficient with respect to the ground truth. The result of the snake segmentation was more promising when compared to the grab-cut's. With a detection accuracy of 76.84% and average

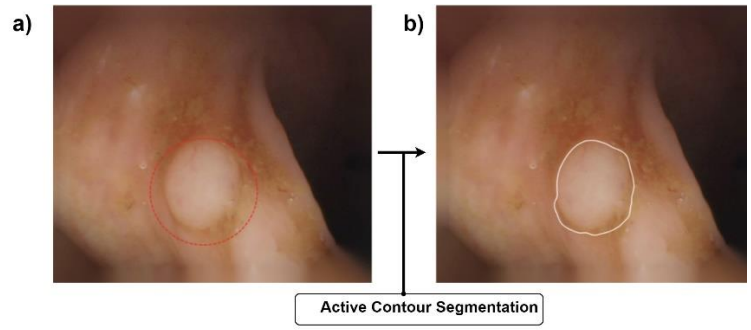

Figure 11. Active contour segmentation. a) Initial seed, b) Snake Contour

segmentation dice score of 42.57%. We observed that the algorithms were failing to segment different cases of polyps, indicating that they could complement each other. Hence the two-stage approach as discussed in the next section.

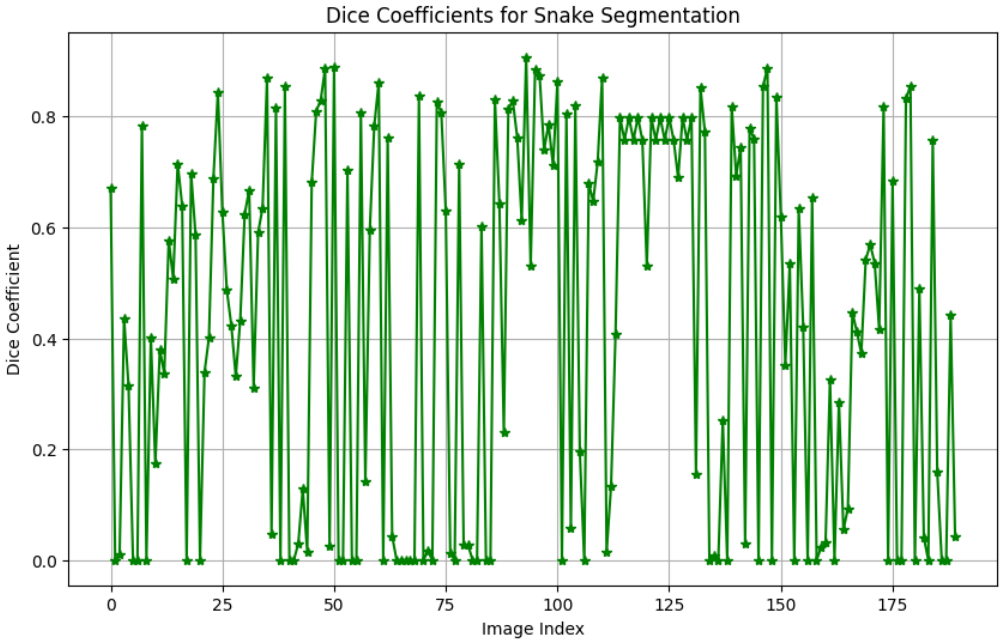

Figure 12. Dice Coefficients of the Snake Segmentation across 190 images

4.1.2.3 The Two Stage Approach

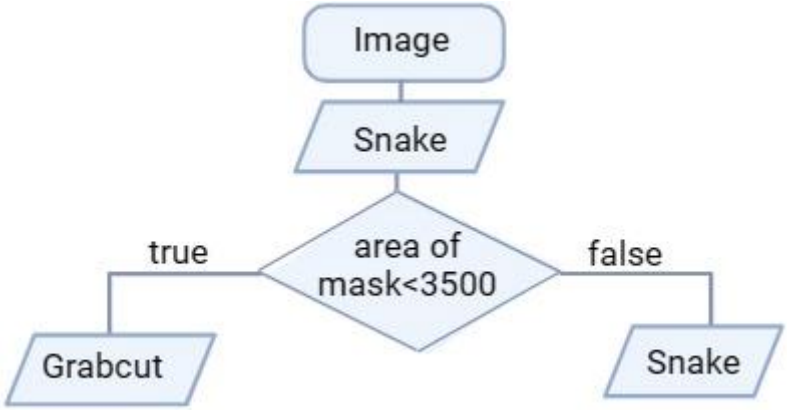

Figure 13. Schema of the two-stage approach

The two segmentation algorithms described above failed to segment all polyp cases. Observing the cases where they failed, we notice that in the grab-cut result, the failure was because:

i. The ROIs extracted do not contain the polyp.
ii. The grab-cut failed to segment the polyp, resulting in edges as segmentation.

In the case of the snake algorithm, it was either the Hough circle detection failed, or the snake contour shrank too small. There wasn't much we could do for the first case while for the latter, we could measure the area of the segmentations and use a threshold to accept or reject the result.

The two-stage approach involves the following:

i. Apply Snake Segmentation and obtain the mask.
ii. Calculate the area of the mask. If the area of the mask is below 3500, we propagate the image to the grab-cut algorithm for segmentation.

The two-stage approach demonstrated an increase in polyp detection and mask segmentation as can be seen in table 2 and figure 14.

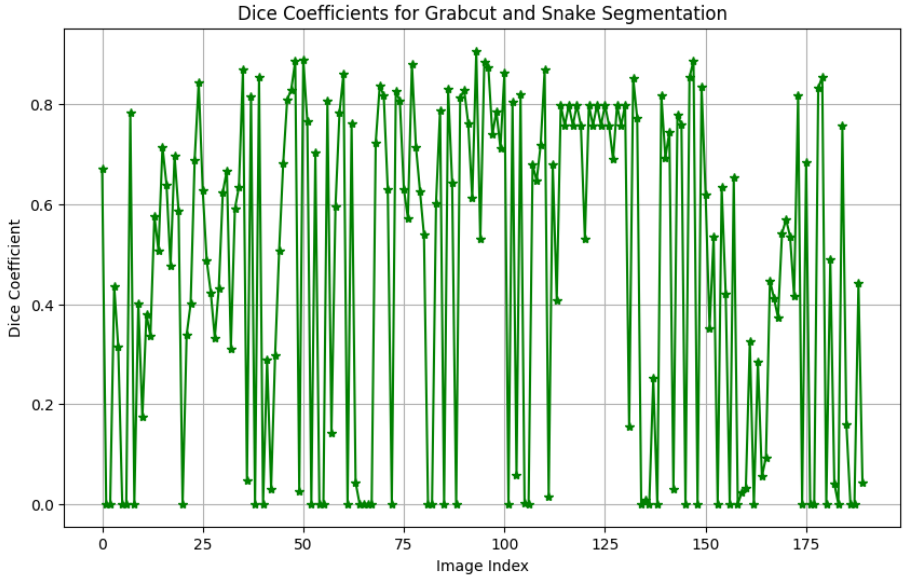

Figure 14. Dice Coefficients of the two Stage Segmentation approach across 190 images

Methods	**Polyp Detection Accuracy (%)**	**Average Polyp Segmentation Dice Score (%)**	**Average Polyp Segmentation Dice Score for detected polyps (%)**
Grab-cut	54.74	29.04	53.10
Snake	76.84	42.57	55.40
Two-stage	79.47	46.67	58.73

Table 2. Metrics obtained from the advanced image analysis pipelines

With peak segmentation coefficient of about 90.72%, we believe that the methods described here have great potential towards specific kinds of polyps with margin for improvements.

4.2 Machine Learning

In this section we will discuss the workflow followed for the Machine Learning system in our experiment. For this part of the task, after pre-processing our dataset, we used two different feature extraction methods to extract the features from our dataset. We then applied machine learning models to classify if the stomach of the patient is normal or it has polyp. Finally, the implemented models were evaluated on the test set to suggest which model performs well for the binary classification of the polyp dataset.

4.2.1 Data

4.2.1.1 Dataset

For the machine learning pipeline, Gastro-vision dataset has been used for training, validation, and testing. The dataset contains images of two classes: Normal stomach and colon polyps. Among the 1789 images of the dataset, we split the dataset as 1351 images for training, 338 images for validation and 100 images for testing. We further used HyperKvasir [22] as another foreign dataset to evaluate our performance of the classifier. The dataset consists of 1028 images that we use for testing purposes only. For both the datasets, the images are divided into two classes only, either normal stomach or colon polyp.

4.2.1.2 Data processing

Before applying feature extraction methods, we preprocessed our training and test dataset by removing green patches in the images. Both training and test datasets were checked if they have any green patches. This was done by checking the average value of the red channel in a region of the image. The green patches were removed by replacing the patch with a specified color (black). After pre-processing the dataset was ready for feature extraction.

4.2.2 Feature Extraction

Feature extraction is a crucial step in the data preprocessing pipeline, aimed at transforming raw data into a format that is suitable for machine learning algorithms. This process involves identifying and selecting relevant features that can enhance the predictive performance of models. Effective feature extraction not only reduces dimensionality but also retains essential information that captures the underlying patterns in the data. Techniques for feature extraction vary depending on the type of data, including statistical methods for tabular data, signal processing methods for time series, and image processing techniques for visual data. Properly extracted features can significantly improve model accuracy and efficiency.

Over the years, a lot of feature extraction methods have been introduced by researchers. We considered Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) as the basis of feature extraction for the machine learning classification part in our project.

4.2.2.1 Gray Level Co-occurrence Matrix (GLCM)

Haralick first created GLCM features in 1973 to characterize image textures for classification [23]. GLCM is a 2D matrix that is used to capture the spatial relationship between pixels in an image. The textual information of an image is identified using this method. The GLCM matrix counts the occurrence of two pixel intensities in an image's region of interest for an angle θ and distance d between pixels. Then it calculates different statistical features from the GLCM

matrix. It takes images as a NumPy array as argument and outputs a data frame of GLCM features. For our experiment, we applied GLCM for our polyp dataset. It first gets the matrix of co-occurrence of pixel intensity pairs at a distance d = 1 [left, right, up, down] and angle, θ = 0 (which is horizontal i.e. left to right. Although there are numerous handcrafted statistical features defined for GLCM, we extracted the features most common for polyp detection namely energy, correlation, contrast, dissimilarity, and homogeneity. The extracted features were then fed into the machine learning model for classification.

4.2.2.1 Local Binary Pattern (LBP)

Ojala [24] introduced the LBP operator in 1996 for texture classification. It has since been implemented for face recognition, facial expression identification, medical imaging, and other applications. It is now commonly used as a feature extractor in many works by researchers. The LBP feature extraction technique uses local texture to create a unique signature for gray-level images. The process begins by calculating the LBP for each pixel in the input image using *N* points circularly distributed at a radius *D* from the center pixel. This operation is performed for all pixels in the image to generate the LBP image. The LBP values obtained are then converted into a one-dimensional array, and a histogram is calculated to represent the distribution of these values. This histogram captures the local texture information in a binary format (sign) and provides a concise summary of the texture characteristics of the image. Adjusting the radius and number of points can make the LBP invariant to multiple scales, making it adaptable for many applications.

The LBP feature extractor algorithm takes images as NumPy array, radius of the circle that is used to sample the points and number of points to sample on the circle as input arguments and outputs a NumPy array containing the LBP features. For this project we took radius =1 and points = 8, so the closest 8-neighbors are used. We employed the 'uniform' method to ensure that the LBP is both grayscale and rotation invariant, enhancing its robustness to variations in lighting and orientation. The extracted features are then fed into different machine learning classification algorithms.

4.2.3 Machine Learning Model Training

Four different machine learning models are implemented in this project for both the feature extraction methods to classify between normal stomach and colon polyp. The dataset is split into train, validation and test set. For both GLCM based feature extraction and LBP based feature extraction the machine learning models used for classification of the dataset are Random Forest, K-Nearest Neighbor (KNN), XGBoost and LightGBM. The best parameters for the aforementioned classifiers are trained using scikit-learn *GridSearchCV* function with 2 folds. The models are then evaluated based on some performance metrics on the test set as well as on a foreign dataset to suggest which model works best for polyp classification.

4.2.3.1 Random Forest

Random Forest (RF) introduced by Breiman [25] in 2001 is a machine learning technique that creates a forest of classification trees based on a bootstrap sample of data. Attributes at each node are randomly selected from a subset of all attributes [26]. RF has gained much popularity as it can handle a wide range of input attributes, including qualitative and quantitative data. Moreover, it assesses the relative relevance of features in classification.

The classifier is declared with the following parameters:

-'n_estimators': [50, 100, 200, 300]

- 'max_depth': [10, 20, 30, 40, 50]

For the GLCM based feature extraction, the best parameters found through grid search method among different options are:

- 'n-estimators' : 200 (the number of trees in the forest)

- 'max_depth' : 50 (the maximum depth of the tree)

For the LBP based feature extraction, the best parameters found through grid search method among different options are:

- 'n-estimators' : 200 (the number of trees in the forest)

- 'max_depth' : 50 (the maximum depth of the tree)

4.2.3.2 K-Nearest Neighbor (KNN) Classifier

KNN is a simple classifier that uses a neural network to assign each pixel to the same class as the training data based on its intensity [27]. It works by identifying the k nearest data points in the training set to a given input based on a distance metric, usually the Euclidean distance. The class of the input is then predicted by a majority vote among its k nearest neighbors. This is a non-parametric, instance-based algorithm, meaning it does not make any assumptions about the underlying data distribution and relies directly on the training data for making predictions. Following parameters are declared for this classifier:

- 'n_neighbors': [3, 5, 7, 9, 11]
- 'weights': ['uniform', 'distance']
-'metric': ['euclidean', 'manhattan']

For the GLCM based feature extraction, the best parameters found for KNN through grid search method among different options are:

- 'metric' : manhattan

- 'n_neighbors' : 11

-'weights': distance

For the LBP based feature extraction, the best parameters found for KNN through grid search method among different options are:

- 'metric' : manhattan

- 'n_neighbors' : 3

-'weights': distance

4.2.3.3 XGBoost

XGBoost (Extreme Gradient Boosting) classifier is a highly efficient and scalable machine learning algorithm based on gradient boosting principles. The XGBoost algorithm differs from simple gradient boosting in that it does not add weak learners sequentially. It's multi-threaded

approach optimizes CPU core utilization for increased speed and performance [28]. XGBoost also incorporates regularization techniques to prevent overfitting and improve generalization, making it robust for various datasets.

The classifier is declared with the following parameters:

-'n_estimators': [50, 100, 200, 300]

- 'max_depth': [10, 20, 30, 40, 50]

4.2.3.4 LightGBM

LightGBM is another popularly known gradient boosting framework that employs tree-based learning techniques [29]. This algorithm utilizes a novel leaf-wise tree growth strategy, which can result in deeper trees and better accuracy compared to level-wise growth strategies. LightGBM has a number of benefits over other existing Gradient Boosting Decision Tree algorithms like faster training speed, higher efficiency, lower memory usage, better accuracy, the ability to handle large-scale data, and support for parallel and GPU learning [29].

Following parameters are declared for this classifier:

-'learning_rate': [0.05, 0.1, 0.2]

-'boosting_type': ['gbdt', 'dart']

-'num_leaves': [31, 50, 100]

4.2.3.5 CatBoost

CatBoost classifier is a robust, high-performance gradient boosting algorithm designed specifically to handle categorical features without extensive preprocessing. During training, it effectively transforms categorical features into numerical values by applying a combination of OGB and ordered TS to handle categorical data [30]. CatBoost also incorporates ordered boosting to mitigate overfitting and improve prediction accuracy.

Following parameters are declared for this classifier:

-'iterations': [50, 100, 200, 300]

-'depth': [4, 6, 8, 10]

4.2.3.6 Support Vector Classifier (SVC)

SVC can efficiently perform linear and non-linear classification through the use of kernel functions, which transform the input data into higher-dimensional space. The algorithm classifies classes using a decision surface based on the most informative points from the training set [31]. SVC is particularly effective for high-dimensional spaces and is robust to overfitting, especially in cases with fewer samples than features.

The classifier is declared with the following parameters:

-'C': [0.1, 1, 10]

-'gamma': [1, 0.1, 0.01, 0.001]

-'kernel': ['rbf']

4.2.4 Experimental Result and Discussion

4.2.4.1 Hyperparameters tuning

To have a better estimation of the performance of the classification models on the test set, the model training and the hyperparameter selection experiments were carried out using GridSearchCV for 2 folds. The best parameters obtained for both GLCM based feature extraction and LBP based feature extraction along with their test and validation accuracy are illustrated in Table 3.

	GLCM Feature Extraction			LBP Feature Extraction		
Models	**Best parameters**	**Validation Accuracy**	**Test Accuracy**	**Best parameters**	**Validation Accuracy**	**Test Accuracy**
Random Forest	max_depth: 20; n_estimators: 300;	0.911	0.920	max_depth: 10; n_estimators: 100;	0.941	0.930
KNN	metric: manhattan; n_neighbors: 11; weights: distance;	0.873	0.890	metric: manhattan; n_neighbors: 3; weights: distance;	0.932	0.930
XGBoost	max_depth: 20; n_estimators: 300;	0.932	0.910	max_depth: 10; n_estimators: 300;	0.956	0.950
LightGBM	boosting_type: gbdt; learning_rate: 0.2; num_leaves: 31;	0.938	0.900	boosting_type: dart; learning_rate: 0.2; num_leaves: 50;	0.956	0.950
SVM	C: 1; gamma: 0.001; kernel: rbf;	0.885	0.930	C: 10; gamma: 1; kernel: rbf;	0.932	0.920
CatBoost	depth: 10; iterations: 300;	0.929	0.930	depth: 8; iterations: 50;	0.953	0.940

Table 3. Hyperparameter Tuning for GLCM and LBP based models

4.2.4.2 Performance Evaluation

The classifier models with the best parameters were implemented on the dataset. Several performance metrics like Accuracy, Precision, Recall and F1 score were calculated for each classifier for both feature extraction methods. We fitted the model on our training data and evaluated the performance metrics on the test data.

	GLCM Feature Extraction			
Models	**Accuracy**	**Precision**	**Recall**	**F1-score**
Random Forest	0.92	0.92	0.92	0.92
KNN	0.90	0.90	0.90	0.90
XGBoost	0.91	0.91	0.91	0.91
LightGBM	0.90	0.90	0.90	0.90
SVM	0.91	0.91	0.91	0.91
CatBoost	0.90	0.90	0.90	0.90

Table 4. Classification Report for GLCM based model on test set

	LBP Feature Extraction			
Models	**Accuracy**	**Precision**	**Recall**	**F1-score**
Random Forest	0.93	0.93	0.93	0.93
KNN	0.94	0.95	0.94	0.94
XGBoost	0.93	0.93	0.93	0.93
LightGBM	0.94	0.95	0.94	0.94

SVM	0.93	0.93	0.93	0.93
CatBoost	0.93	0.93	0.93	0.93

Table 5. Classification Report for LBP based model on test set

The best performing models (LightGBM) were retrained for both feature extraction cases to obtain their best model metrics. Moreover, for the best classifier, their Receiving Operating Characteristic (ROC) curve and their confusion matrix were also computed.

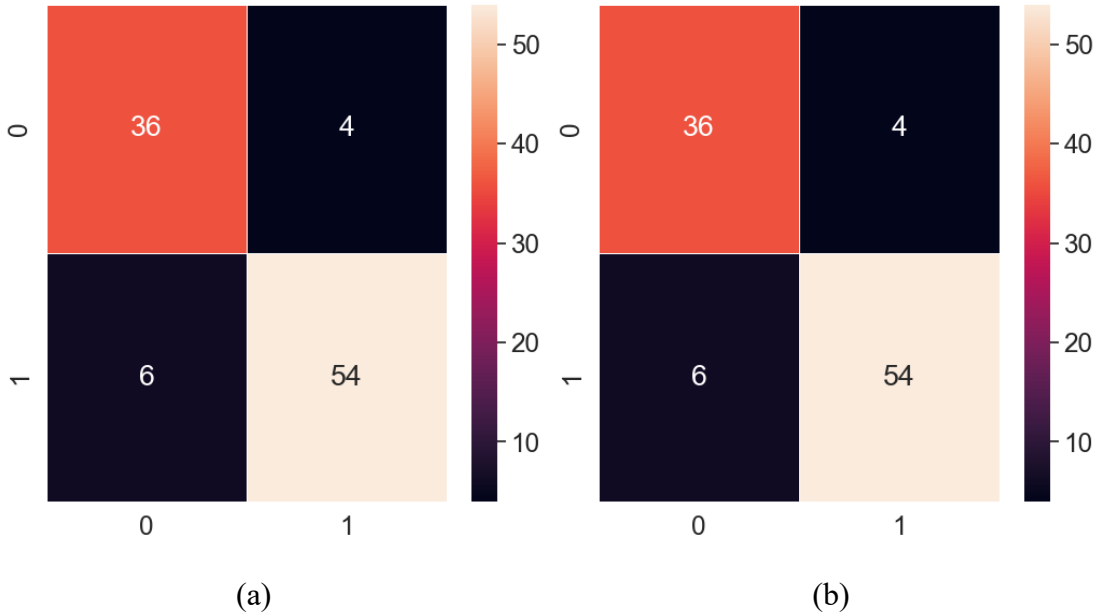

Figure 15. Confusion Matrix for (a) GLCM based model and (b) LBP based model

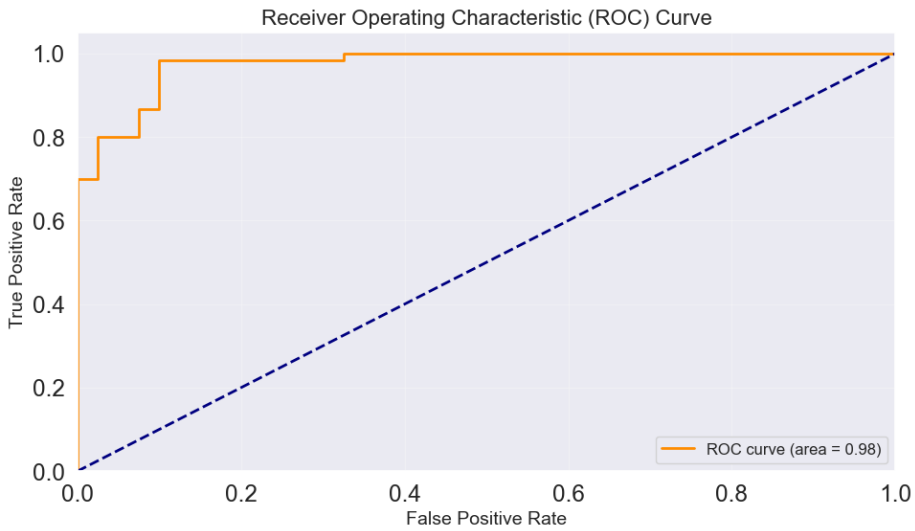

Figure 16. ROC Curve for GLCM based model

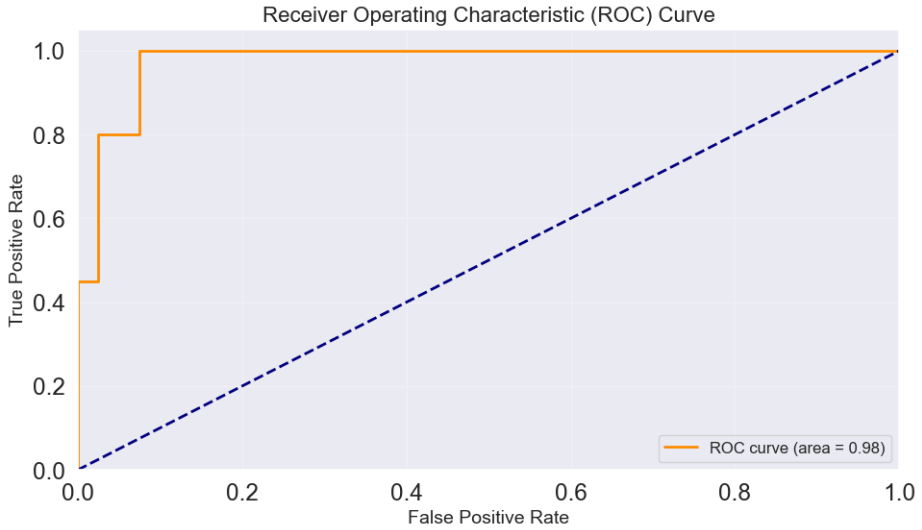

Figure 17. ROC Curve for LBP based model

For further evaluation of our machine learning based polyp classification project, the best performing LightGBM model was also tested on another foreign dataset (HyperKvasir dataset).

	GLCM Feature Extraction				LBP Feature extraction			
Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Precision**	**Recall**	**F1-score**
LightGBM	0.72	1.0	0.71	0.83	0.90	1.0	0.90	0.94

Table 6. Classification Report for HyperKvasir test set

The performance was again evaluated using the same performance metrics used before. The results obtained are satisfactory for both feature extraction methods. For the GLCM based feature extraction method, using the LightGBM classifier on the foreign dataset almost 72% accuracy is achieved, whereas for the LBP based feature extraction method LightGBM classifier gives 90% accuracy. With the obtained results based on our experiment we can conclude that for the machine learning classification pipeline the best choice would be to use LBP as a feature extractor and LightGBM classifier for accurate polyp classification.

4.3 Deep Learning

4.3.1 Deep Learning for Polyp Segmentation

In this section, we discuss the development and evaluation of deep learning models for colon polyp segmentation, a critical task for improving colorectal cancer detection and diagnosis.

We employ five state-of-the-art neural network architectures, three of which are based on the U-Net architecture, namely: U-Net++ with a ResNet34 backbone, U-Net++ with an EfficientNetB0 backbone, and U-Net with an EfficientNetB0 backbone. Each model leverages advanced convolutional neural network (CNN) backbones to enhance feature extraction and segmentation performance. The choice of the combinations of the architectures and backbones was to have a relative comparison of the U-Net and U-Net++ performance and have an idea of the performance of ResNet34 relative to the EfffficientNetB0 backbone. We use the same data and training hyperparameter configuration in the three experiments. In another set of experiments, we implemented and tested the two most recent and state of the art models in colon polyp

segmentation. The two models namely, DuckNet [32] and Metapolyp [33] are also discussed in the next sections.

As mentioned earlier, we combined CVC-ColonDB and Etis-LaribPolypDB datasets to obtain a total of 576 samples which were split into the training and validation set using 80:20 ratio. We maintained the CVC-ClinicDB as the test set. The training set was augmented using four methods namely, random rotation, brightness, zoom and horizontal flip. This augmentation increased the training set by 400% which is necessary because deep models feed on data and the better the data the better the model's generalization and robustness.

For the model hyperparameters, we use the default model configuration with input shape of 256x256 pixels. The training hyperparameters were however chosen as follows:

i. Optimizer: Adam, with learning rate set to 1e-3 and Cosine annealing schedular(T_max=50, eta_min=1e-6).
ii. Batch size: 32 samples.
iii. Loss: Dice Loss.
iv. Metrics: Dice Coefficient.
v. Max Epoch: 100, with early stopping criteria of 15 epoch patience and validation loss monitoring.
vi. We save only the best model using the check pointer.

The next section discusses the model architectures and results of the experiments.

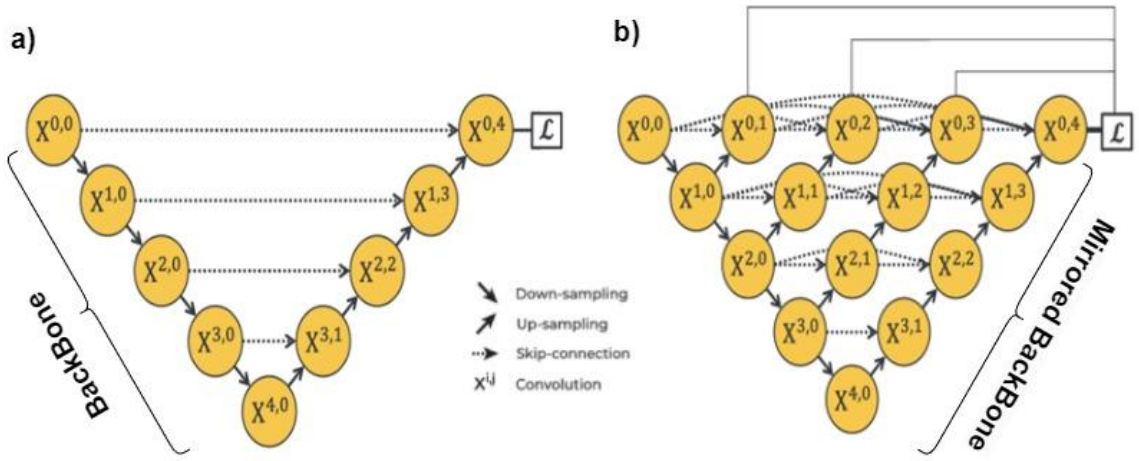

Figure 18. Segmentation Architectures a) U-Net, b) U-Net++

4.3.1.1 U-Net with EfficientNetB0 Backbone

The U-Net architecture is a widely used convolutional network for biomedical image segmentation, known for its U-shaped structure that facilitates effective feature extraction and localization. In this configuration, we use EfficientNetB0 as the backbone network.

EfficientNetB0 Backbone: EfficientNetB0 is a highly efficient convolutional neural network that balances network depth, width, and resolution using a compound scaling method. It achieves superior performance with fewer parameters and computational resources compared to traditional CNNs.

U-Net: The U-Net architecture consists of an encoder-decoder structure with skip connections that directly connect corresponding layers in the encoder and decoder paths. This design helps preserve spatial information and improves segmentation accuracy. By incorporating EfficientNetB0, the model benefits from advanced feature extraction, enhancing its ability to segment polyps accurately.

We trained the model from scratch using the configuration mentioned earlier and evaluated the model on the hold-out test set. The training curve is as shown in fig 19.

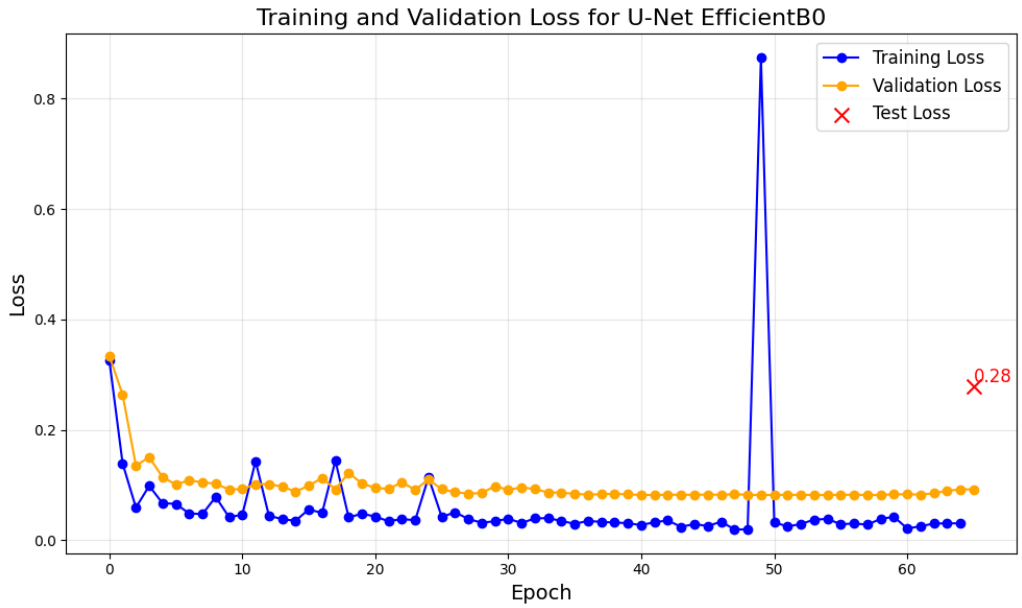

Figure 19. Training and Validation Loss for U-Net EfficientNetB0

The figure shows how smooth the training was, evidently validating the correct choice of hyperparameters. Both the training and validation losses decrease rapidly, indicating effective learning. We noticed the sudden significant spike in the training loss around epoch 50, which suggests a momentary instability or an anomaly during training. This could be due to a variety of reasons such as a temporary learning rate increase, batch anomalies, or data inconsistencies. The validation loss curve does not exhibit the same spike as the training loss around epoch 50, indicating that the spike in training loss was likely an isolated event that did not affect the overall generalization performance.

After the spike, the training loss returns to a lower value and remains relatively stable with slight fluctuations while the validation loss maintained its course. Overall, it exhibits a general downward trend, which is typical of a well-performing model. We saved the best performing model based on the validation loss of 0.08. The test loss with a value of 0.28, being close to the validation loss indicates that the model has generalized well to new, unseen data.

4.3.1.2 U-Net++ with EfficientNetB0 Backbone

U-Net++ is an advanced variant of the original U-Net architecture designed for image segmentation tasks. It introduces a series of nested, densely connected convolutional layers, which enhance the network's ability to capture fine-grained features and improve segmentation accuracy. This model combines the U-Net++ architecture with the EfficientNetB0 backbone, leveraging the strengths of both networks to achieve high-performance segmentation. This

structure allows for better segmentation of complex and small structures, such as polyps in colonoscopy images.

EfficientNetB0 Backbone: In this model, EfficientNetB0 serves as the backbone, providing robust feature extraction capabilities while maintaining computational efficiency. We also trained this configuration using the same parameters as mentioned in the earlier experiment and evaluated the model on the hold-out test set. The training curve is as shown in fig 20.

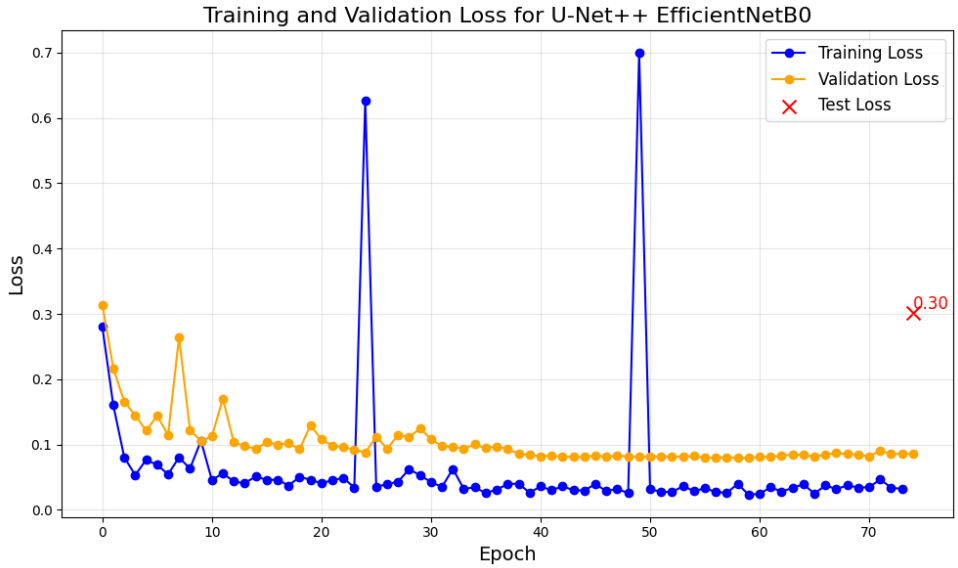

Figure 20. Training and Validation Loss for U-Net++ EfficientNetB0

We observed from the figures that both training and validation losses decrease rapidly in the initial epochs, indicating effective initial learning. After the initial decrease, the validation loss remains relatively stable, showing minor fluctuations and indicating good generalization. The significant spikes in the training loss around epochs 25 and 50 suggest temporary instabilities or anomalies during training. However, these did not affect the validation loss significantly. The test loss of 0.30, being a little bit higher than that of the U-net model, suggests that the U-net was more effective on the task and data used.

4.3.1.3 U-Net++ with ResNet34 Backbone

U-Net++: As described previously, U-Net++ enhances the traditional U-Net architecture with dense skip connections, which facilitate better feature reuse and improve segmentation accuracy. In this configuration, we use ResNet34 as the backbone network.

ResNet34 Backbone: ResNet34 is a residual network that uses skip connections to mitigate the vanishing gradient problem, enabling deeper network training. It consists of 34 layers, including convolutional, batch normalization, and activation layers, and is known for its efficiency and strong performance in feature extraction.

The final configuration was also trained with same parameters as the last two experiments, we evaluated the model on the hold-out test set and observed the following. The training curve is as shown in fig 21.

We observed similar training curve and spikes as indicated in the last two experiments; this spike most likely indicates some inconsistency in the training data. The test loss like that of the U-net++ with EfficientNetB0 backbone indicates a good performance of the configuration.

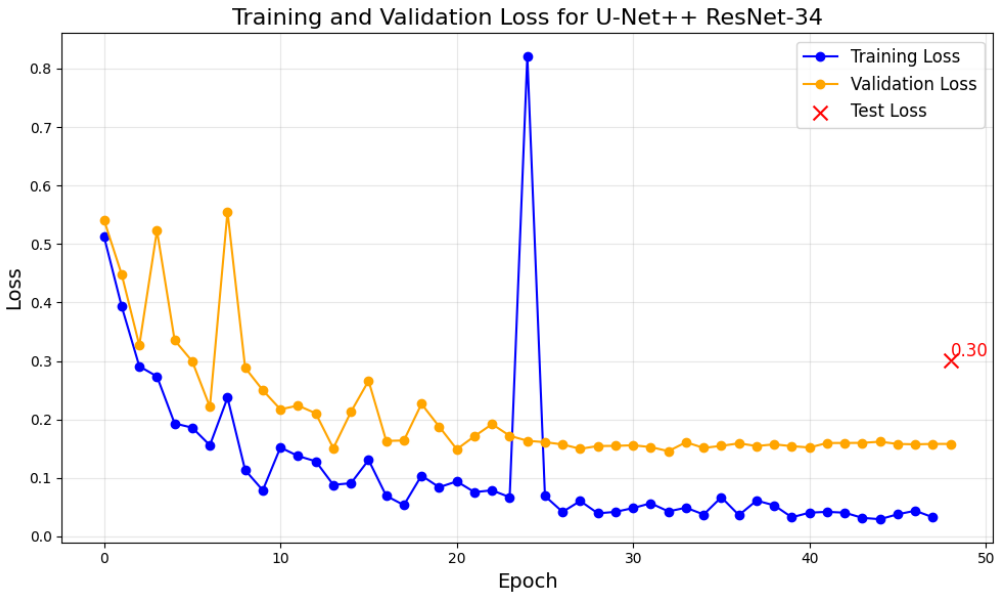

Figure 21. Training and Validation Loss for U-Net++ ResNet34

4.3.1.4 Meta-Polyp

Meta-Polyp is an advanced polyp segmentation model that utilizes a unique integration of MetaFormer and U-Net architectures, combined with novel upsampling techniques to enhance segmentation quality. The model features a dual-stage encoder that begins with ConvFormer blocks for capturing significant local features, transitioning to Transformer blocks in later stages to integrate global contextual information. This design allows the model to extract multi-resolution features efficiently. In the decoder part, Meta-Polyp employs a multi-scale upsampling block with a specific "Level-up Upsampling" technique, designed to improve texture and detail in the upsampled images. The final segmentation mask is produced by merging these upsampled features followed by a 1x1 convolution that narrows down the feature maps to a single filter.

The architecture introduces significant innovations like the ConvFormer block, which merges local feature extraction with global context through a series of depthwise and separable convolutions, normalization, and skip connections. Additionally, the Transformer blocks in the encoder focus on capturing long-range dependencies within the input data. The model's novel "Level-up Upsampling" technique in the decoder is specifically designed to tackle the common issue of texture loss during traditional upsampling, enhancing texture quality effectively. Moreover, the multi-scale upsampling block reiterates feature map processing multiple times at each decoder step, ensuring fine details are preserved and overall segmentation quality is improved. These innovative elements make Meta-Polyp highly efficient in processing and potentially very effective in clinical applications, providing a robust tool for accurate polyp segmentation.

We used the same data configuration as in the case of the U-Net based models. The performance of Meta-Polyp was evaluated using several metrics on the test dataset, as detailed below in Test Data Metrics graph. These metrics demonstrate the model's strong performance in accurately segmenting polyps, with a high Dice Coefficient, and relatively low loss values.

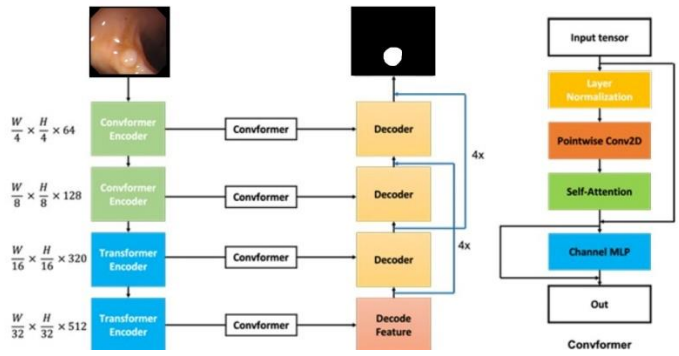

Figure 22. Meta-polyp Architecture

The results show the model's ability to maintain high segmentation accuracy despite the challenges posed by out-of-distribution datasets and small polyps.

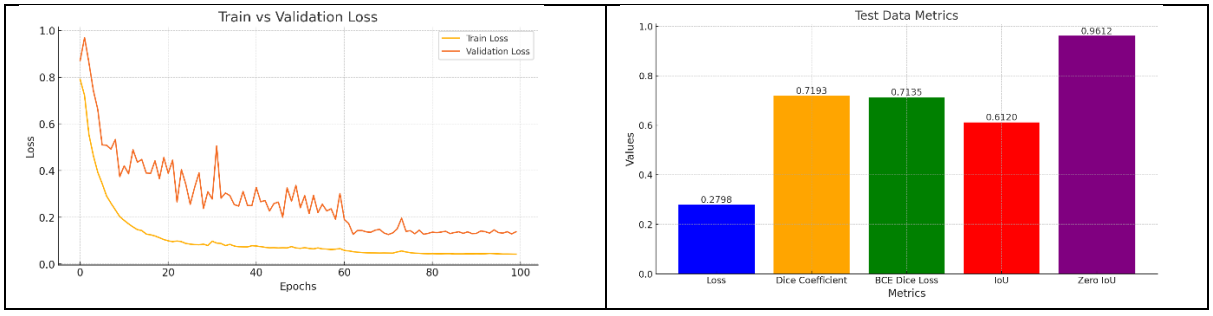

Figure 23. Training and Validation Loss with Test Metrics for Meta-Polyp

4.3.1.5 DUCK-Net

DUCK-Net is a sophisticated supervised convolutional neural network specifically designed for the task of polyp segmentation in medical imaging. It adopts an encoder-decoder framework, which is advantageous for segmentation as it allows the model to capture both high-level context and detailed spatial information efficiently. The encoder features a novel convolutional block named DUCK (Deep Understanding Convolutional Kernel), which comprises six different convolutional layer variations running in parallel. These include the Residual Block for fine detail capture, Midscope and Widescope Blocks using dilated convolutions to enhance the receptive field for mid and high-level feature capture, a Separated Block that employs 1xN and Nx1 convolutions, a Residual Downsampling mechanism to preserve original image information, and a Secondary Downscaling Layer that helps in retaining low-level details without direct processing.

The decoder of DUCK-Net mirrors the encoder's structure but focuses on upscaling features to match the original image resolution, utilizing skip connections to regain spatial details lost during downsampling. Notably, DUCK-Net employs addition rather than concatenation to

merge feature maps from the encoder and decoder, optimizing memory usage and computational efficiency. This network introduces several innovative components like the custom DUCK Block which integrates various convolutional operations to enhance feature selection at each stage, thus improving the accuracy in locating polyp boundaries. Additionally, the model's effective use of residual and dilated convolutions allows it to capture a wide range of contextual details and maintain spatial accuracy across dimensions.

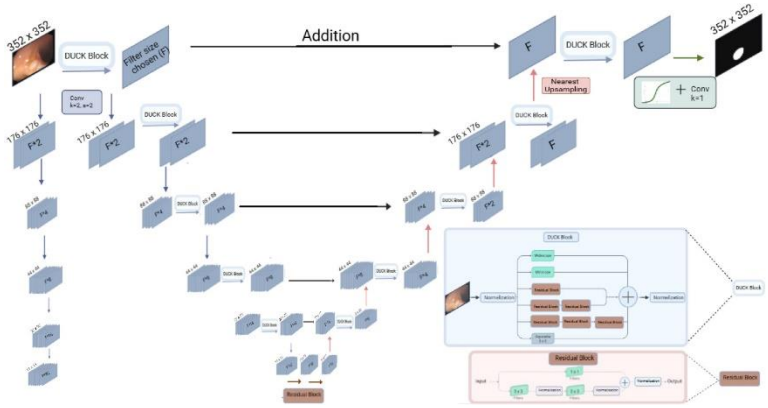

Figure 24. DUCK-Net Architecture

The performance of DUCK-Net was evaluated, and the curve was obtained.

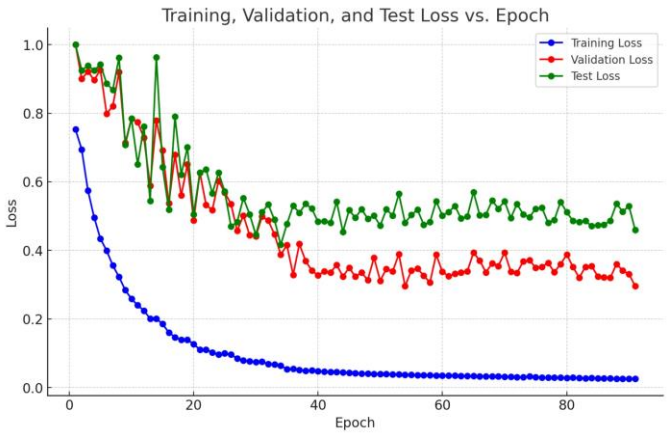

Figure 25. Training, Validation and Test loss for Duck-Net

4.3.1.4 Experimental Results and Discussion

Comparing all the experiments, we obtain the following table:

Model Config	Trainable Parameter	Epoch	Train Loss	Validation Loss	Test Loss
U-net & EfficientNetB0	**6.3M**	49	0.0307	0.0923	**0.2787**
U-net++ & EfficientNetB0	6.6M	58	0.0316	**0.0853**	0.3009
U-net++ & ResNet34	26.1M	**32**	0.0326	0.1581	0.3006
Meta-polyp	28.7M	100	0.0427	0.1386	0.2798
Duck-Net	38.9M	91	0.0255	0.2961	0.4595

Table 7. Train and test results for deep learning models

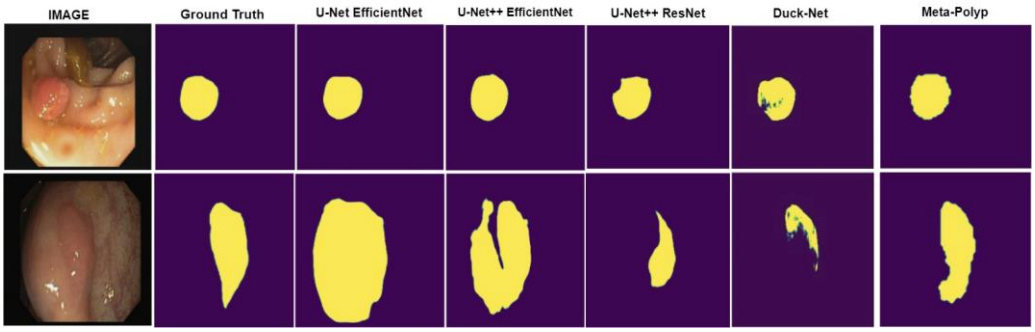

Figure 26. Sample Prediction Results

Best Performance: The "U-net & EfficientNetB0" model has the best performance on the test set (lowest test loss) despite having the fewest trainable parameters.

Generalization: The "U-net++ & EfficientNetB0" model, while having the lowest validation loss, does not generalize as well to the test set compared to the "U-net & EfficientNetB0."

Model Complexity: Increasing model complexity with "U-net++ & ResNet34" or "Meta-polyp" does not necessarily lead to better performance; in fact, "U-net++ & ResNet34" has one of the highest validation loss and comparable test loss to the simpler "U-net++ & EfficientNetB0." The DUCK-Net model, however, shows a clear sign of overfitting, perhaps this could have been as a result of low data for a model of its complexity or the difference in train and test set data distribution.

4.3.2 Deep Learning for Polyp Classification

4.3.2.1 Data Preparation

In comparison to the machine learning models obtained using GLCM and LBP features, we trained four deep learning models on the gastro-vision dataset following these procedures:

I. We split the training data into two sets, 80% for training and 20% for validation.
II. We maintained the 100 images for testing as used during the machine learning phase.
III. All images were resized to 224 by 224 to match the default input of the models.
IV. We applied data augmentation techniques. i.e. RandomHorizontalFlip, ColorJitter, and Random Rotation.

4.3.2.2 ResNet-18 Models

The ResNet-18 model comprises of 18 trainable layers, which include the initial convolution layer, the convolutional layers within the residual blocks, and the fully connected layer.

1. We trained a ResNet-18 model from scratch using the following parameters:
   a. Number of trainable layers: all
   b. Loss function: Cross entropy loss
   c. Optimizer: Adam(lr=1e-3), with Cosine Annealing (T_max=50, eta_min=1e-6).
   d. Max epoch: 100, with early stopping criteria
2. We fine-tuned a ResNet-18 model using the same sets of parameters above except that the number of trainable layers was set to 7.

4.3.2.3 EfficientNetB0 Models

EfficientNetB0 is composed of several building blocks with convolutional layers, batch normalization layers, and activation functions. The main components include stem convolution layer, Multiple Mobile Inverted Bottleneck Convolution (MBConv) blocks with different configurations, final 1x1 convolution layer, and fully connected (dense) layer.

1. We trained an EfficientNetB0 model using the same set of parameters as used during the full training of the ResNet-18 model.
2. We also fine-tuned an EfficientNetB0 model specifying the number of trainable layers to seven (7), just as in the case of the ResNet-18 model.

4.3.2.4 Experimental Results and Discussion

The training losses in all scenarios decrease over time and stabilize at a low value, indicating the model is learning effectively from the training data. The validation losses however show high variability in 3 scenarios (both cases of the ResNet18 model and EfficientNetB0 full training), with several spikes and drops, but does not decrease steadily.

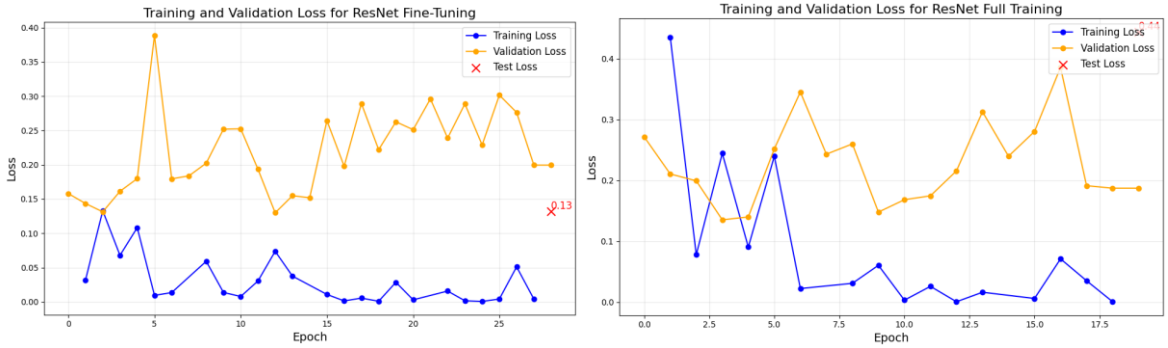

Figure 27. Training and Validation Loss for ResNet18

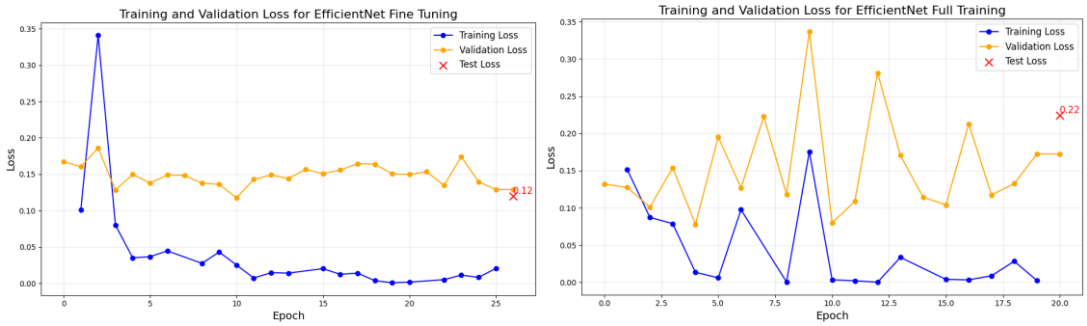

Figure 28. Training and Validation Loss for EfficientNetB0

This suggests issues with overfitting or instability in the validation performance. In the case of the EfficientNetB0 fine tuning, the validation loss maintained a steady course, indicating little overfitting but superior performance when compared to the other 3 cases. The test loss in both fine tuning cases appears lower than full training cases, indicating poorer generalization of the full training to the test data. Overall, the fine tuning of EfficientNetB0 model shows better test performance and indicates the least training approach adjustments to address validation instability and overfitting concerns.

5. Conclusion

In this work, we implemented and evaluated 5 different pipelines for automatic classification and segmentation of colon polyps. The first pipeline, based on the grab-cut algorithm consisted of multiple advanced processing analysis for the segmentation of polyps. The early stages of the pipeline included the removal of specular reflections and black masks via image inpainting, followed by the grab-cut implementation and finally a tailored post-processing step based on morphological operations. The second pipeline, also an image processing approach for the segmentation of polyps, was based on a fine-tuned active contour method. We adapted the preprocessing phase of the previous pipeline and adopted novel methods for delineating polyp structures to enhance the detection. The third one, involved a double feature extraction method, one based on gray level cooccurrence matrix and the other based on local binary pattern. We utilize several machine learning algorithms to classify polyp images based on the extracted features. In the fourth pipeline, we implemented multiple CNN-based deep learning models to compare the results of the handcrafted feature in the machine learning pipeline. Finally, we compared several deep learning networks to automatically segment colon polyps. For each pipeline several experiments were carried out, to find the best set of hyperparameters considering each pipeline's different stages and methodologies. As shown in previous sections, we thoroughly evaluated the performance of these methods (segmentation and classification separately), comparing intra-pipeline variants and finally all the pipelines in an independent hold out test set. Concisely, the results showed that U-Net with EfficientNetB0 backbone led to the best performance compared with the other segmentation pipelines and model, achieving a Dice-Coefficient of 0.7213. On comparing the classification results, we observed that the fine-tuned EfficientNetB0(pretrained on ImageNet) model outperformed the others. Our work highlights some of the key aspects of deep convolutional neural networks: their ability to extract meaningful features and hierarchical representations useful for different computer vision tasks. In our work we successfully trained CNNs to classify and segment whole images containing polyps, forcing the network to adjust its internal weights to identify the discriminative characteristics of the polyps. Our work also demonstrates the effectiveness of Efficient Net models on these tasks. It is no coincidence that the EfficientNetB0 model performs the best on both segmentation and Classification of the polyps. By reutilizing the pretrained models, we were able to improve the model's convergence and achieve better results, proving that pretrained weights were still useful even with different objective tasks. This well contrasts with the highly sensitive and detailed handcrafting of the feature extraction pipelines that were needed to get decent results in the machine learning based classification and image processing-based segmentation.

References

[1] Wu, Zhenyu, et al. "Colorectal Polyp Segmentation in the Deep Learning Era: A Comprehensive Survey." arXiv preprint arXiv:2401.11734 (2024).

[2] Dheir, Ibtesam M., and Samy S. Abu-Naser. "Classification of anomalies in gastrointestinal tract using deep learning." (2022).

[3] Khan, Mehshan Ahmed, et al. "Gastrointestinal diseases segmentation and classification based on duo-deep architectures." Pattern Recognition Letters 131 (2020): 193-204.

[4] De Lange, Thomas, Pål Halvorsen, and Michael Riegler. "Methodology to develop machine learning algorithms to improve performance in gastrointestinal endoscopy." World journal of gastroenterology 24.45 (2018): 5057.

[5] Gross, Sebastian, et al. "Polyp segmentation in NBI colonoscopy." Bildverarbeitung für die Medizin 2009: Algorithmen—Systeme—Anwendungen Proceedings des Workshops vom 22. bis 25. März 2009 in Heidelberg. Springer Berlin Heidelberg, 2009.

[6] Afify, Heba M., Kamel K. Mohammed, and Aboul Ella Hassanien. "An improved framework for polyp image segmentation based on SegNet architecture." International Journal of Imaging Systems and Technology 31.3 (2021): 1741-1751.

[7] Jha, Debesh, et al. "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning." Ieee Access 9 (2021): 40496-40510.

[8] Galdran, Adrian, Gustavo Carneiro, and Miguel A. González Ballester. "Double encoder-decoder networks for gastrointestinal polyp segmentation." Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I. Springer International Publishing, 2021.

[9] Liu, Fangjin, et al. "Dbmf: Dual branch multiscale feature fusion network for polyp segmentation." Computers in Biology and Medicine 151 (2022): 106304.

[10] Sanderson, Edward, and Bogdan J. Matuszewski. "FCN-transformer feature fusion for polyp segmentation." Annual conference on medical image understanding and analysis. Cham: Springer International Publishing, 2022.

[11] Fan, Deng-Ping, et al. "Pranet: Parallel reverse attention network for polyp segmentation." International conference on medical image computing and computer-assisted intervention. Cham: Springer International Publishing, 2020.

[12] Shen, Yutian, Xiao Jia, and Max Q-H. Meng. "Hrenet: A hard region enhancement network for polyp segmentation." Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24. Springer International Publishing, 2021.

[13] Xu, Ziheng, et al. "Temporal correlation network for video polyp segmentation." 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2022.

[14] Chen, Mingzhu, et al. "Single-modality endoscopic polyp segmentation via random color reversal synthesis and two-branched learning." 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2022.

[15] Tajbakhsh, Nima, Suryakanth R. Gurudu, and Jianming Liang. "Automated polyp detection in colonoscopy videos using shape and context information." IEEE transactions on medical imaging 35.2 (2015): 630-644.

[16] Bernal, Jorge, et al. "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians." Computerized medical imaging and graphics 43 (2015): 99-111.

[17] https://polyp.grand-challenge.org/ETISLarib/

[18] Jha, Debesh, et al. "Gastrovision: A multi-class endoscopy image dataset for computer aided gastrointestinal disease detection." Workshop on Machine Learning for Multimodal Healthcare Data. Cham: Springer Nature Switzerland, 2023.

[19] Telea, Alexandru. "An image inpainting technique based on the fast marching method." Journal of graphics tools 9.1 (2004): 23-34.

[20] Yang, Daipeng, et al. "An overview of edge and object contour detection." Neurocomputing 488 (2022): 470-493.

[21] Bernal, Jorge, Javier Sánchez, and Fernando Vilarino. "Integration of valley orientation distribution for polyp region identification in colonoscopy." Abdominal Imaging. Computational and Clinical Applications: Third International Workshop, Held in Conjunction with MICCAI 2011, Toronto, ON, Canada, September 18, 2011, Revised Selected Papers 3. Springer Berlin Heidelberg, 2012.

[22] Borgli, Hanna, et al. "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy." Scientific data 7.1 (2020): 283.

[23] Haralick, Robert M., Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification." IEEE Transactions on systems, man, and cybernetics 6 (1973): 610-621.

[24] Yang, Zhiguang, and Haizhou Ai. "Demographic classification with local binary patterns." Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Proceedings. Springer Berlin Heidelberg, 2007.

[25] Katzmann, Alexander, et al. "Deep random forests for small sample size prediction with medical imaging data." 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, 2020.

[26] Nedjar, Imane, et al. "Random forest based classification of medical x-ray images using a genetic algorithm for feature selection." Journal of Mechanics in Medicine and Biology 15.02 (2015): 1540025.

[27] Ramteke, R. J., and Khachane Y. Monali. "Automatic medical image classification and abnormality detection using k-nearest neighbour." International Journal of Advanced Computer Research 2.4 (2012): 190.

[28] Ramraj, Santhanam, et al. "Experimenting XGBoost algorithm for prediction and classification of different datasets." International Journal of Control Theory and Applications 9.40 (2016): 651-662.

[29] Rufo, Derara Duba, et al. "Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM)." Diagnostics 11.9 (2021): 1714.

[30] Samat, Alim, et al. "GPU-accelerated CatBoost-forest for hyperspectral image classification via parallelized mRMR ensemble subspace feature selection." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021): 3200-3214.

[31] Rahman, Md Mahmudur, Bipin C. Desai, and Prabir Bhattacharya. "Medical image retrieval with probabilistic multi-class support vector machine classifiers and adaptive similarity fusion." Computerized Medical Imaging and Graphics 32.2 (2008): 95-108.

[32] Dumitru, Razvan-Gabriel, Darius Peteleaza, and Catalin Craciun. "Using DUCK-Net for polyp image segmentation." Scientific reports 13.1 (2023): 9803.

[33] Trinh, Quoc-Huy. "Meta-Polyp: a baseline for efficient Polyp segmentation." 2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2023.