People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
KASDI MERBAH UNIVERSITY - OUARGLA
Faculty of New Technologies of Information and Telecommunication
Department of Computer Science and Information Technology

MASTER

Domain : Computer Science
Field : Artificial Intelligence and Data Science
Submitted by : Tahar Rezzag Bedida and Abdeldjalil Hammouya

Thesis :

Improving SAM model for medical image segmentation

Evaluation Date : 24/06/2024

Before the Jury :

Dr.Oussama Aiadi	MCA	Supervisor	UKM Ouargla
Dr.Khadra Bouanane	MCB	Examiner	UKM Ouargla
Dr.Merabti Hocine	MCA	Jury Chair	UKM Ouargla

Academic year: 2023/2024

Acknowledgements

We thank first and foremost Allah the Almighty the greatest of all, for His guidance and facilitation in completing this thesis and we hope that it will benefit us in our religion and our worldly life.

We would like to extend our thanks and gratitude to our dear Dr Oussama Aiadi, our supervisor, who aided us with his practical and invaluable guidance, insightful knowledge, and supportive encouragement throughout this research journey. His insightful feedback and belief in us have been instrumental in shaping this work. We are truly blessed to have had the opportunity to work under his supervision.

Our deep thanks go to our parents, families, and siblings, for their support, unconditional love, patience, and understanding. We are forever indebted to them, their continuous moral support has been the driving force behind this thesis.

We want to extend our gratitude to our respected teachers, professors, and faculty members at Kasdi Merbah University's Department of Computer Science and Information Technology for all the support and for providing the academic environment, special thanks to our professors and their teachings have not only enriched our intellectual growth but have also shaped us into better human beings.

We are grateful for the opportunity to study alongside our esteemed colleagues and fellow students, their cooperation, constructive feedback, and insightful discussions we shared, which have undoubtedly contributed to the refinement of this research.

Before the end, we wish to express our profound gratitude once more to each individual for their immense support, encouragement, significant contribution, and unconditional love, Words alone cannot describe the depth of our appreciation for all that you have done to our life.

In the end, may Allah (SWT) accept this humble effort as a good deed and grant us all success in this life and the hereafter.

Tahar Rezzag Bedida
Abdeldjalil Hammouya

Abstract

Early detection of polyps in the colon is crucial for preventing colorectal cancer, the second leading cause of cancer-related deaths globally. However, accurate identification of polyps can be challenging due to factors like subtle visual cues, variable lighting, and human fatigue. This work aims to adapt the Segment Anything Model (SAM) to segment colonoscopy polyp by replacing its encoder with a lightweight convolutional neural network. Additionally, we strive to enhance the model's accuracy and automation through the implementation of zero-shot learning. This approach involves utilizing a pre-trained object detection model with K-means clustering algorithm to extract the bounding box prompt, which serves as auxiliary information for SAM, thereby improving its performance on unseen polyp data without the need for fine-tuning or manual prompt design. The proposed method reduces the number of SAM encoder parameters from 91M to 3M. It demonstrates superior performance compared to some existing approaches that work to fine-tune SAM with large number of parameters.This work offers a contribution to computer-aided polyp detection. It paves the way for more efficient and accurate polyp segmentation systems, ultimately improving early cancer diagnosis and patient care.

Keywords: Medical Imaging, Polyps Segmentation, Segment Anything Model (SAM),Few-Shot Learning (FSL) Zero-Shot Learning (ZSL), Vision Transformers (ViTs), Convolutional Neural Network (CNN)

ملخص

يعد الكشف المبكر عن الأورام في القولون أمرًا بالغ الأهمية للوقاية من سرطان القولون والمستقيم السبب الرئيسي الثاني للوفيات المرتبطة بالسرطان على مستوى العالم. ومع ذلك، تحديد هذه الاورام بشكل دقيق يشكل تحديًا بسبب عوامل مثل الإشارات المرئية الدقيقة والإضاءة المتغيرة و التعب البشري. يهدف هذا العمل إلى تكييف نموذج تجزئة أي شيء (SAM) على تجزئة أورام القولون عن طريق استبدال المشفر الخاص به (Encoder) بشبكة عصبية تلافيفية (CNN) خفيفة الوزن. بالإضافة إلى ذلك، حاولنا تعزيز دقة النموذج وأتمتته من خلال تنفيذ التعلم صفري اللقطات (zero-shot learning) . وذلك باستخدام نموذج مدرب مسبقًا للكشف عن الكائنات مع خوارزمية تجميع لاستخراج المربع المحيط بالورم، والذي يستخدم كمعلومة مساعدة ل SAM ، وبالتالي تحسين أدائه على البيانات غير المرئية بدون الحاجة إلى الضبط الدقيق أو التحديد اليدوي. الطريقة المقترحة تقلل من عدد معلمات تشفير SAM 91M من 3M پ إلى وتحقق أداء متفوق مقارنة ببعض الأساليب الحالية التي تعمل على ضبط SAM بعدد كبير من المعلمات. طريقتنا تحقق أداءً جيدًا في تجزئة السلائل في مجموعة بيانات Kvasir-SEG ، حيث حققت IoU 64.7% و Dice 78.4% و IoU 70% و Dice 81% على التوالي بدون استخدام المطالبات (prompts) وباستخدام المطالبات. يقدم هذا العمل مساهمة في الكشف عن اورام القولون بمساعدة الكمبيوتر. إنه يمهد الطريق لمزيد من أنظمة تجزئة السلائل ذات كفاءة ودقة، مما يؤدي في نهاية المطاف إلى تحسين التشخيص المبكر للسرطان ورعاية المرضى.

الكلمات المفتاحية: التصوير الطبي، تجزئة اورام، نموذج تجزئة أي شيء (SAM) ،التعلم بقليل من اللقطات التعلم باللقطات القليلة (Few-Shot Learning) ،التعلم باللقطات الصفرية (Zero-Shot Learning) ، Vision Transformers (ViTs) ،الشبكة العصبية التلافيفية مدرب مسبقًا (CNN)

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Introduction

The field of artificial intelligence (AI) has witnessed remarkable progress in recent years, revolutionizing various aspects of our lives. AI includes a set of technologies that enable, machines to learn, solve problems, and make decisions based on a set of data, making it a powerful tool for tackling many real-world problems that were previously intractable.

Deep learning (DL) is one of the most advanced and contributing sub-fields of artificial intelligence, which uses artificial neural networks to mimic the structure and function of the human brain. DL models have achieved impressive performance in a wide range of tasks by extracting complex patterns and relationships from large data sets, which was difficult to extract using handcrafted methods [29, 30], and it has revolutionized natural language processing areas such as translation, speech recognition, sentiment analysis, text generation and summarization, and in computer vision there has also been great progress in image and video recognition, such as object detection, image classification, and segmentation. This led to progress in other fields such as robotics, agriculture, and health care.

The ability to accurately diagnose diseases is critical to effective patient care. However, traditional diagnostic methods can be time-consuming, subjective, and prone to human error. AI-based systems provide a compelling opportunity to enhance healthcare delivery, by assisting in the diagnosis of diseases, image analysis, and segmentation of medical images aim to automatically identify specific areas of interest within images to improve the accuracy, speed, and efficiency of diagnosis.

In the medical field, obtaining labeled data is expensive, takes a long time, and requires specialized expertise, especially when dealing with rare diseases or specific medical conditions, which hinders traditional deep-learning models from performing optimally because of their need for huge amounts of labeled training data. This is what led to the emergence of few-shot learning (FSL), which is an emerging subfield in artificial intelligence that aims to enable machines to learn effectively from limited amounts of data.

One area where medical image segmentation can be of assistance is the detection and analysis of gastrointestinal (GI) polyps. These are abnormal growths in colon tissue, some of which develop on the lining of the digestive system and become cancerous over time, early detection and removal of polyps is crucial to preventing the development of colorectal cancer, which is the second leading cause of cancer-related deaths globally [31]. This thesis aims to contribute to the development of a more powerful, lightweight, and efficient deep-learning model for diagnosing gastrointestinal polyps in the digestive system based on few-shot learning, which would help

medical professionals in the early detection of colon cancer and improve patient care.

1.2 Problematic

Meta AI's Segment Anything Model (SAM) [26] is a zero-shot learning image segmentation model trained on the SA-1B dataset, capable of adapting to new image distributions and tasks. It showed amazing performance in generating valid segmentation masks from different prompts such as spatial or textual clues, outperforming most fully supervised models.

Despite its good results in many fields, directly applying the pre-trained SAM to medical image segmentation does not yield satisfactory performance as shown in the figure 1.1, due to the significant domain shift between natural images, which SAM was trained on, and medical images, which makes SAM's performance may be constrained by its capacity to generalize to new datasets and tasks without requiring substantial fine-tuning [32].

Also, fine-tuning SAM for specialized domains like medical imaging faces many challenges. SAM's large encoder, which is based on a Vision Transformer (ViT) architecture, contains a substantial number of parameters (91M in the smallest version ViT-B) which makes it computationally expensive and memory-intensive to finetune the entire encoder on limited-data domains like medical imaging [33], and the large encoder capacity of SAM increases the risk of overfitting, leading to poor generalization performance [1]. In addition, SAM relies heavily on prompts and is affected by errors in them [34], and struggles with segmenting specific objects autonomously, as it relies on manual user input prompts like points or bounding boxes to identify targeted objects. This manual intervention may cause the model to encounter difficulties and make errors in determining correct segmentation masks and predict impractical and inefficient masks, especially in medical imaging where automatic segmentation of specific anatomical structures is required and precise segmentation is crucial.

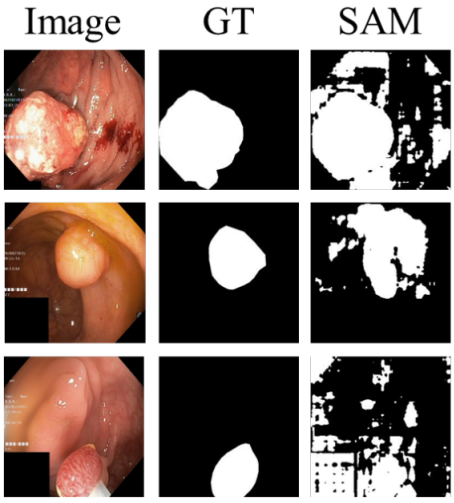


Figure 1.1: Examples of SAM ViT-B results in polyps segmentation[1]

1.3 Overview on the related techniques

There are numerous methods for segmenting polyps in medical images, broadly categorized into traditional, machine learning (ML)-based, and deep learning (DL)-based approaches.

- **Traditional methods**
  - **Region-based methods:** These techniques group pixels with similar characteristics (e.g., color, intensity) to form a mask around the polyp like region growing [35].
  - **Edge-based methods:** These methods rely on edge detection algorithms like Canny edge detection [36] to identify object boundaries based on sharp intensity changes.
  - **Threshold-based methods:** These methods segment objects based on intensity thresholds such as Otsu threshold [37]. Global and local thresholding are the two main subcategories of threshold methods, global thresholding applies a single threshold to the entire image, while local thresholding adapts the threshold based on image regions.
  - **Watershed-based methods:** These methods treat the image as a topographic surface and use watersheds to separate objects based on intensity valleys [38].

  The figure 1.2 shown the traditional methods used for image segmentation:

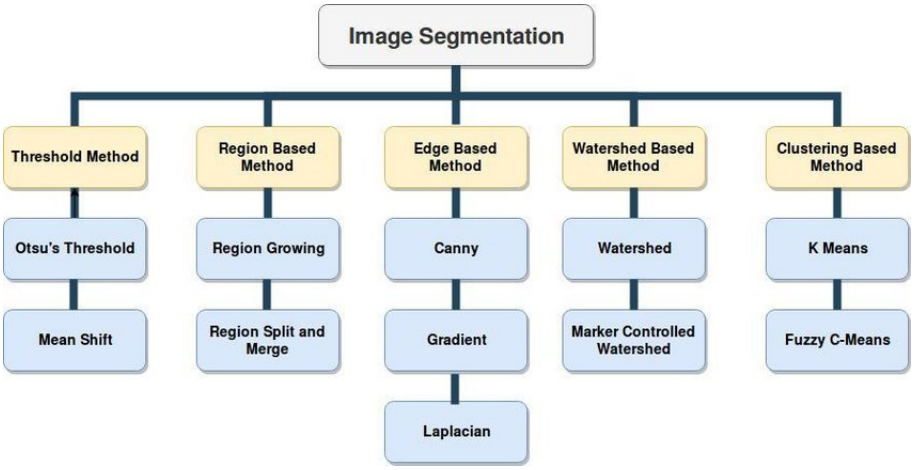


Figure 1.2: Traditional Image Segmentation Methods[2]

While this methods useful and effective for simple objects, they susceptible to noise, struggle with complex shapes and requires careful parameter tuning.

- **ML-based methods**

  ML-based methods like k-nearest neighbors (KNN) [39], support vector machines (SVM) [40, 41] and random forests [42], and clustering like K-Means clustering [43] and Fuzzy C-Means [44], was also used in segmentation literature to separate target objects from the background and faced challenges including robust to outliers, needing for data word-processing, features engineering and requires careful parameters tuning [45, 46].

- **DL-based methods** Popular DL-based methods for image segmentation are based on conventional neural networks which capturing spatial relationships like U-Net [17] and DeepLab [47] or Vision Transformer which capture long-range dependencies like SAM [26] or hybrid which leveraging the strengths of both CNN and ViTs like U-Net transformer model (UNETR) [21]. These methods require significant computational resources and large datasets for training.

1.4 Motivation

Lightweight models are essential for real-time applications on resource-constrained devices, we aims to significantly reduce the SAM model's size and computational resources without sacrificing accuracy. That enables faster inference times, making the model more suitable for deployment in clinical settings and facilitating efficient polyp analysis during colonoscopies.

Zero-shot learning is particularly valuable in medical imaging where labeled data is often scarce and expensive to acquire. It allows the model to learn from unseen polyp classes by transferring knowledge from similar labeled classes. This broadens the model's generalization capabilities and can enables robust even for rare cases.

Self-prompting is another good technique that injects knowledge into the model during training automatically, by strategically prompting the model with spatial hints or descriptions related to polyp segmentation, that helps to learn more effective feature representations and improve its ability to differentiate polyps from healthy tissue. This leads to more precise segmentation and fast convergence and leads to a reduction in human errors that occur due to fatigue and lack of experience.

Our work proposes a SAM architecture that combines zero-shot learning, self-prompting, and lightweighting to improve the accuracy and efficiency of SAM for polyp segmentation, and opens doors for more robust and faster polyp detection, ultimately contributing to earlier cancer diagnosis and improved patient outcomes.

1.5 Overview on the proposed method

- We developed a lightweight variant of the SAM model. This variant replaces the computationally expensive vision transformers with a small-depth convolutional neural network in the encoder.

- We propose a self-promoting approach that eliminates manual intervention errors caused by fatigue and inefficiency when analyzing large volumes of diverse-quality images. This method tackles the challenge of limited medical image data by leveraging zero-shot learning properties. It utilizes an object detection model to extract informative prompts (bounding boxes) based on a textual description of the target polyp. Furthermore, we introduce a k-means aggregation strategy to generate a more accurate bounding box by combining outputs from the object detection model. This self-prompting approach ensures precise and rapid segmentation of specific anatomical structures.

1.6 Thesis Structure

The rest of this thesis is organized as follows:

In the second chapter, we will provide an overview of artificial intelligence and deep learning, focusing on medical image segmentation. Then we will discuss the segment anything model and some work related to our work that aimed at adapting SAM to the medical image segmentation.

In the third chapter, we will present our proposed method in detail, starting with the proposed encoder, passing through the method of extracting prompts, to the mask decoder that predicts the final mask.

After that, in the fourth chapter, we will describe the dataset used and its pre-processing with the evaluation metrics used to evaluate the results. Then we will take a look at the work

environment and the experiments tested and discuss the obtained results.

Finally, in the fifth chapter, we will provide a general conclusion of the thesis, where we will present the main result of the work, its importance and limitations, and what can be achieved in future works.

Chapter 2

Work Background

2.1 Introduction

In recent decades, the rapid advancement of computational hardware and the massive increase in the data available, impacted the progression of artificial intelligence (AI) to become a successful approach to solving complex problems across various industries such as healthcare, finance, and retail. AI is a branch of computer science that focuses on creating techniques, agents, and systems that can solve certain tasks as humans would. It involves analyzing large volumes of data for efficient pattern identification and then making predictions or decisions [48]. In medical image analysis, AI can reduce human errors in disease detection and automate efficiently the analysis leading to faster diagnoses and treatment planning, which makes the healthcare industry more efficient and accurate. Segment Anything Model (SAM) is a cutting-edge deep learning approach designed to segment any object in an image based on user prompts and generate masks corresponding objects. Applying SAM to medical images presents hurdles, due to the unique complexities, variations and data scarcity, requiring the model to adapt to unseen scenarios [49, 50].

In this chapter, we provide fundamental concepts in machine learning and deep learning that represent the background of our work. Additionally, we discuss medical image analysis with details in segmentation task including the existing works. We also introduce the segment anything model (SAM) architecture and few-shot learning, which used to help SAM in handle unseen images for more accurate prompting to solve its challenges in polyp segmentation.

2.2 Computer Vision

Computer Vision (CV) is a field of artificial intelligence (AI) that is concerned with enabling computers and systems to extract from digital images/videos its content known as image features, such as shape, illumination, and color then utilize this meaningful information in AI system to make decisions. In other words, CV aims to enable computers to 'see' and understand the visual world, similar to human visual perception.[51]

2.2.1 Computer Vision Applications:

The Current widespread integration of computer vision (CV) techniques in a wide variety of real-world applications is significantly improving overall human life quality. Some key as-

pects of computer vision include image recognition, object detection, image segmentation, facial recognition, motion analysis, and machine vision.

The most famous computer vision application is self-driving cars, which is used to identify the correct path, traffic signals, pedestrians, and obstacles on the road in order to make the correct decision in real time. Retail checkout is another application of CV, it is used to track the customer's behavior inside the store and automatically identify and calculate the purchased products for payment.

There are also many other applications such as surveillance and security to enhances safety measures in various settings, facial recognition, agricultural monitoring, manufacturing quality control and augmented reality.

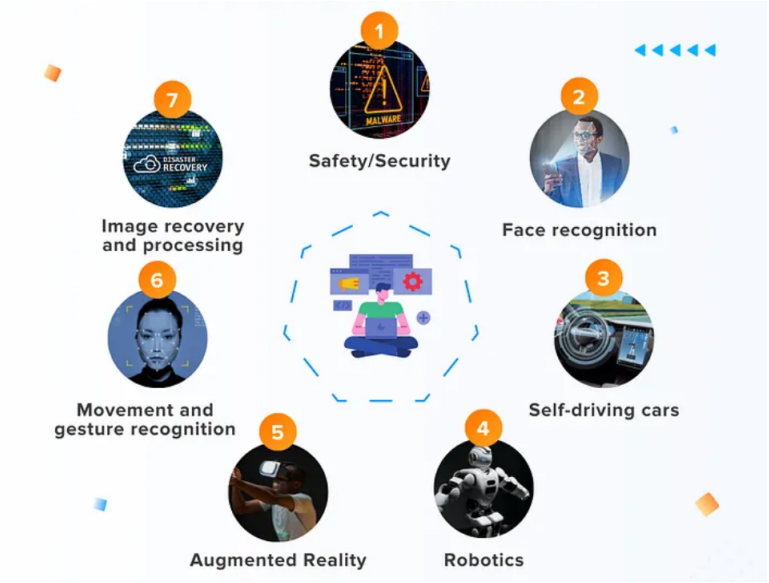


Figure 2.1: Computer vision application [3]

2.2.2 Medical Image Analysis:

Medical image analysis plays a crucial role in disease understanding, clinical diagnosis, and treatment planning. It is the process of extracting meaningful information from different medical image modalities, such as x-rays, ultrasound, microscopy, magnetic resonance imaging (MRI), computed tomography (CT), and nuclear imaging (PET and SPECT), using various computational techniques. This process involves several steps, including image preprocessing, feature extraction, classification, and segmentation. Medical image analysis has witnessed significant progress and promising results in recent years with advancements in deep learning and it is used in various applications, such as counting and identifying cells in a microscopy image, detecting cancerous anomalies in the cells, removing inconsistencies due to human error, segmenting tumor tissues from necrosis.[52]

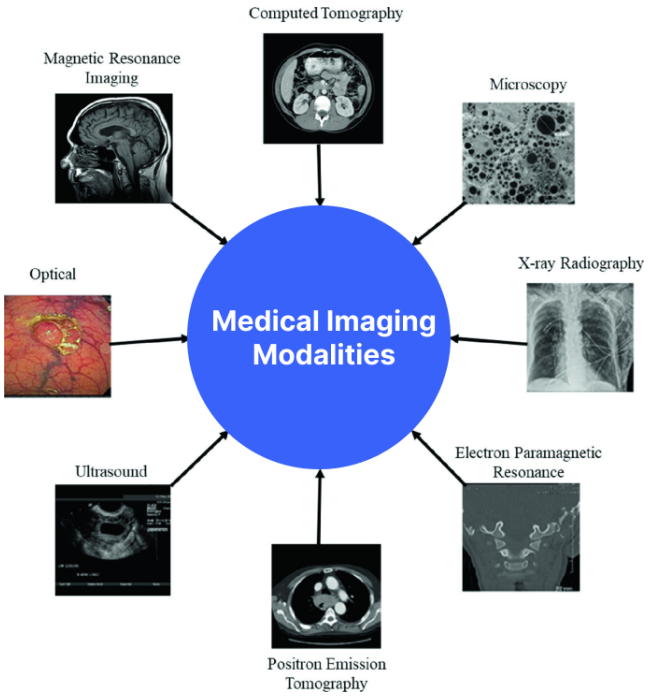


Figure 2.2: Medical image modalities [4]

2.3 Machine Learning (ML)

Machine learning is a field of artificial intelligence that focuses on the development of algorithms and models based on statistics that allow computers to identify patterns from experience and make predictions or decisions with minimal human intervention and without the need for explicit programming [53]. Each machine-learning model has a particular learning pattern when trained with data, below we explained some existing learning patterns:

2.3.1 Machine Learning Paradigms:

2.3.1.1 Supervised Learning:

In supervised learning, the algorithms are trained using labeled data in the form of input-output pairs $(\boldsymbol{x}, \boldsymbol{y})$ to determine the best mapping function $\boldsymbol{f}$ that maps the inputs to the outputs $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ in which the inputs are attributes or features that are related to the output and the outputs are the target or label of interest that we want the algorithm to learn to predict it [54]. Regression and classification.are the two main subcategories of supervised learning.

2.3.1.2 Unsupervised Learning:

In unsupervised learning the training data is unlabeled (we have only the input features), and the methods in these fields try to explore the data and find some hidden structure and relationships within, without needing any supervision or prior knowledge about the outcome (labels) [55]. Commonly use this learning technique for clustering such as K-means (based on centroid) and DBSCAN (based on density), dimensionality reduction like principal component analysis(PCA), and singular value decomposition(SVD).

2.3.1.3 Reinforcement Learning:

In this paradigm, the learning system is called an agent that can observe the environment and interact with it, by selecting and then performing actions resulting in receiving rewards or penalties (punishment) with a new state that the agent will act upon it. The agent learns to take actions that maximize rewards over time in a given state. [56].

2.3.1.4 Transfer Learning (TL):

Transfer learning in machine learning is a technique of re-utilizing the knowledge gained in a machine learning model from the source domain in a given task to improve learning for another target task in a different but related target domain. It is a common technique used when facing the data scarcity problem.

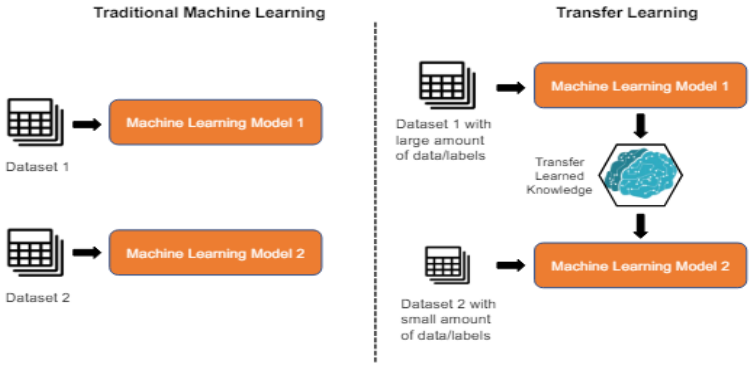


Figure 2.3: The difference between traditional ML and TL [5]

2.3.1.5 Few-Shot Learning (FSL):

In many real-world scenarios obtaining such datasets can be challenging due to various constraints such as scarce disease, privacy concerns, and ethical considerations [57]. Few-shot learning is defined as the process of learning from experience $\boldsymbol{E}$ with 'prior knowledge' that contains one or few labeled instances related to task $\boldsymbol{T}$ while being evaluated by a performance measure ($\boldsymbol{P}$) that assesses the learning procedure [57]. FSL is closely related to knowledge transfer where a model is previously trained on large data then it is used in similar tasks with fewer training data.

it is often described as an N-way-K-shot problem, where train set $\boldsymbol{D_{train}}$ the model previously trained on, and the support set $\boldsymbol{D_{support}}$ (auxiliary dataset) used in training the new task and the query set $\boldsymbol{q}$ that represents as a test set for which to predicted.

$$\boldsymbol{D_{support}} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})_{\boldsymbol{i-1}}^{\boldsymbol{K}}\}_{\boldsymbol{i-1}}^{\boldsymbol{N}} \tag{2.3.1}$$

where:

$\boldsymbol{N}$ is the number of classes each class has a $\boldsymbol{K}$ annotated sample.

When the number of samples is K >1, it represents a few-shot learning setting. In contrast, when K = 1, it is considered a one-shot learning setting. The last scenario occurs when K = 0, indicating the absence of training samples. This setting is the extreme case of few-shot learning (FSL). In zero-shot learning the model works by associating observed and non-observed classes

through some form of auxiliary information this can be textual descriptions, attributes, or other forms of semantic knowledge.

2.4 Deep Learning Concepts

Deep learning (DL) is a subfield of Machine Learning that focuses on the development of networks that handle different data types such as numeric, image, text and audio intending to learn data representation with multiple levels of abstraction, unlike the machine learning algorithms which often require handcrafted data features (representation), deep learning networks that can automatically discover patterns and representations allowing them to solve complex tasks.[55]

The core of deep learning is artificial neural networks that draw inspiration from the interconnected structure of biological neurons of the human brain. They have a great capability to approximate any function to desired level of accuracy, given enough hidden layers and neurons.

The Remarkable abilities of deep learning networks to process patterns with accuracy on a level with the human brain have been demonstrated. This has led to their implementation in a variety of domains and tasks, including natural language processing for tasks like speech recognition and computer vision for tasks such as object detection and image classification.

2.4.1 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) are computational models inspired by the structure and function of biological neural networks in the human brain. The history of ANNs dates back to McCulloch and Pitts's work in 1943 they were motivated by the biological neuron that consists of :

- **Dendrites:** these act as receivers that collect the input signals.
- **Soma:** a neuron cell body that processes the input signals.
- **Axon:** it is the transmitter of the output of this neuron.
- **Synapse:** The point of connection to the other neurons.

In the biological nervous system of the brain has a cosmic number of neurons around $\mathbf{10^{11}}$(100 billion). Neurons perceive messages from other neurons via connections between synapses and dendrites. When a neuron receives enough signals, it fires a processed electrical impulse from the soma, which then travels along its axon and stimulates other neurons connected to it. this procedure establishes the foundation of neural networks in the brain [58].

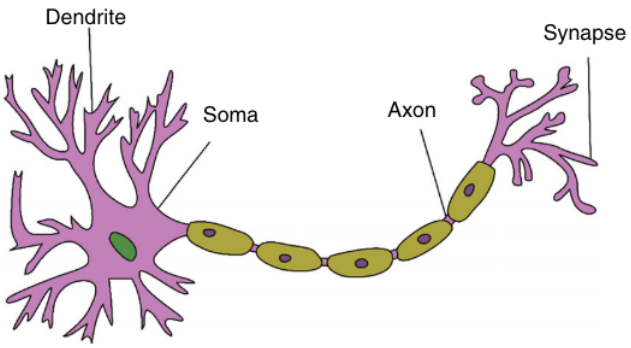


Figure 2.4: Biological Neuron [6]

Similarly McCulloch and Pitts introduced a computational model of a single-neuron(Perceptron) that replicates the work of biological neurons by combining multiple inputs $\boldsymbol{x_1, ...x_i}$ in weight sum where each input is multiplied by parameter $\boldsymbol{w_i}$ called weight, an offset parameter $\boldsymbol{w_0}$ (corresponds to the firing threshold in biological neuron) is added to the weighted sum that is represented as follow:

$$\boldsymbol{a = \sum_{i=1}^{d} w_i x_i + w_0} \tag{2.4.1}$$

A non-linear activation function $\boldsymbol{g(.)}$ is applied to the result of the weighted sum so that $\boldsymbol{z = g(a)}$. non-linear activation functions are employed to facilitate the creation of intricate mappings between the network's inputs and outputs. This capability is essential for learning and modeling complex data types such as images, video, audio, as well as non-linear or high-dimensional datasets.

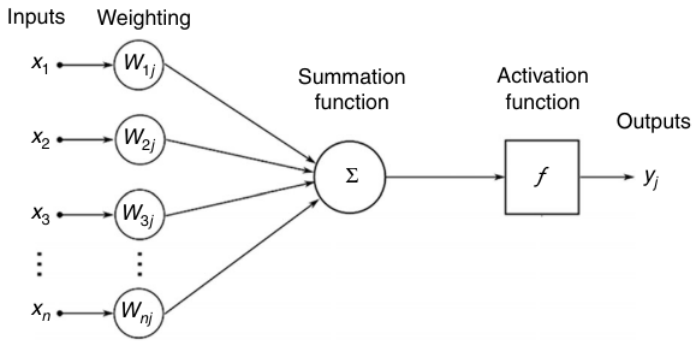


Figure 2.5: Artificial Neuron [6]

Some common non-linear function :

- **Sigmoid:** it is a mathematical operation that maps real input values to a restricted range between zero and one. Its mathematical representation is in:

$$\boldsymbol{g(x) = \frac{1}{1 + e^{-x}}} \tag{2.4.2}$$

- **Tanh:** Similar to sigmoid, Tanh maps the input values between -1 and 1. The Thanh function is defined as:

$$g(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.4.3}$$

- **RELU:** RELU, short for rectified linear unit, is an activation function that ensures all input values are positive. It works by setting negative inputs to zero and leaving positive inputs unchanged. It is defined as:

$$g(x) = max(0, x) \tag{2.4.4}$$

an artificial neural network is composed of 3 connected parts: an input layer, one or more hidden layers, and an output layer with each layer having a set of perceptrons as shown in Figure

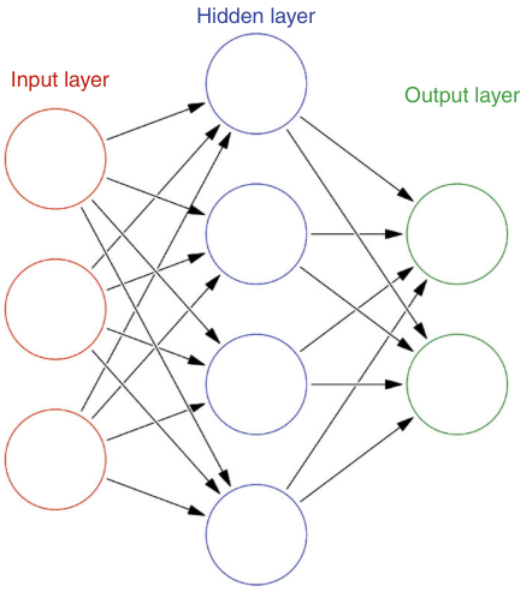


Figure 2.6: Artificial Neural Network [6]

where each layer has an arbitrary number of neurons, and all layers are concatenated with connections between them with every connection having associated weight. The learning process in a neural network starts with a neuron, gets input from other neurons, and then performs a mathematical operation (activation), it passes the result to other neurons in the next layer. This process continues until the final layer, where the output is compared to a ground truth value using a loss function.

To improve the network's performance, an optimization step called back-propagation is used. Back-propagation updates the connections between neurons based on the error (the difference between the predicted output and the ground truth) to increase the network's ability to produce accurate results.

The loss function is a crucial component in training neural networks, as it measures the difference between the predicted output and the true output, guiding the optimization process to find the optimal set of weights. Different loss functions can be used in training neural network depending on the specific problem and desired outcome include:

- **Mean Absolute Error (MAE):** MAE is a loss function used in regression tasks to measure the average magnitude of errors in a set of predictions, without considering their direction, it takes the average of the absolute differences between the predicted values and the actual values [59].

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i|} \tag{2.4.5}$$

where:

$\boldsymbol{y_i}$: The actual value

$\boldsymbol{\widehat{y}_i}$: The predicted value

$\boldsymbol{n}$: The total number of data

- **Mean Squared Error (MSE):** MSE is a loss function used in regression tasks that quantifies the magnitude of the error which is the average of the squared differences between the actual and the predicted values [59]. The mean square error is defined as:

$$\boldsymbol{MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \tag{2.4.6}$$

- **Binary Cross-Entropy Loss (BCE):** BCE or log loss is loss function is a commonly used for binary classification problems. It quantify the randomness between the predicted probability of a class and the actual value, it is calculated from the negative value of the summation in the logarithm value of the probabilities of the predictions against the total number of data samples [60].

$$\boldsymbol{BCE = -\sum_{i=1}^{n} y_i \log(\widehat{y}_i) + (1 - y_i)\log(1 - \widehat{y}_i)} \tag{2.4.7}$$

- **Cross-Entropy Loss for Segmentation:** Cross-entropy loss is a common loss function used in semantic segmentation tasks [60], It is employed to measure the difference between the dissimilarity between the predicted and ground truth segmentation maps pixel by-pixel and it is calculated using the following formula:

$$\boldsymbol{segmentationCE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\widehat{y}_i)} \tag{2.4.8}$$

where:

$\boldsymbol{N}$: The total number of pixels in the image

$\boldsymbol{C}$: The number of classes

$\boldsymbol{y_i}$: The ground truth segmentation map

$\boldsymbol{\widehat{y}_i}$: The predicted segmentation map

2.4.2 Convolutions Neural Networks(CNNs)

Convolutional Neural Networks (CNNs) are a class of deep neural networks that are behind various computer vision tasks like image recognition, object detection, and image classification. CNNs are designed typically to process data that has a grid-like topology, such as images, which can be considered as a 2D grid of pixels.[55]

The great success of CNN is behind the ability to learn the complex representation of visual data using a series of layers, each layer learns different levels of patterns such as edges and textures, which are then combined in deeper layers to recognize more sophisticated features and objects, enabling machines to have vision similar to humans. Figure 2.7 shows key components of basic CNN architecture

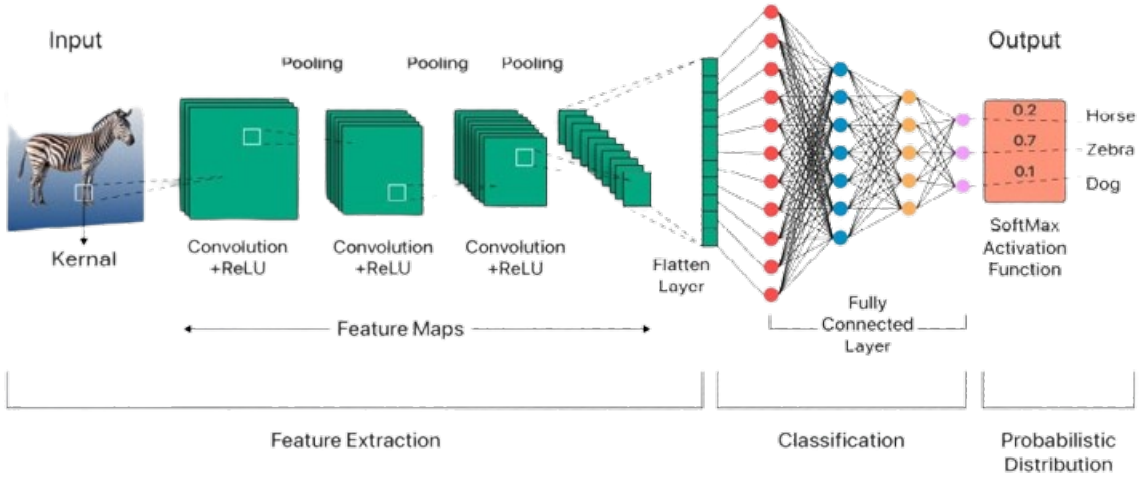


Figure 2.7: CNN architecture for image classification [7]

CNNs consist of key components that enable them to process data effectively,this components include:

2.4.2.1 Convolution layer:

First an input x of size $m \times m \times r$ where m represents the height and width and r is the depth for example in an RGB image the depth is three. A Set of k filters (also known as Kernels) available in each convolutions layer where a filter W has $n \times n \times q$ dimensions here n must be smaller than m and q is either equal to or smaller than r with each k filter is associated with a bias b.

The convolution is a process of sliding a filter over the image as shown in figure 2.17 and applying element-wise multiplication between image patch and filter weights then summing the result into a single output pixel (dot product) and bias is added as provided in equation 2.4.9 that represents the convolution operation. The result of the convolution layer is k feature maps $h^{\boldsymbol{k}}$ which captures the presence of a particular pattern or feature in the input data.[61]

$$h^{\boldsymbol{k}} = W^{\boldsymbol{k}} * x + b^{\boldsymbol{k}} \tag{2.4.9}$$

The size of feature map is adjusted by certain parameters which are:

- **Stride (S):** Stride is a parameter that modifies the speed (step size) of the filter sliding over the input Matrix.
- **Input size (M):** is the width and height of input x of convolution layer.
- **Padding (P):** padding refers to adding layers of zero pixels to the image, ensuring uniform processing when applying filters and other operations.
- **Filter size (F):** is the size of filter in convolution layer.

All these parameters affect the final feature map size resulting in:
size of feature map $= \frac{M-F+2P}{S} + 1$

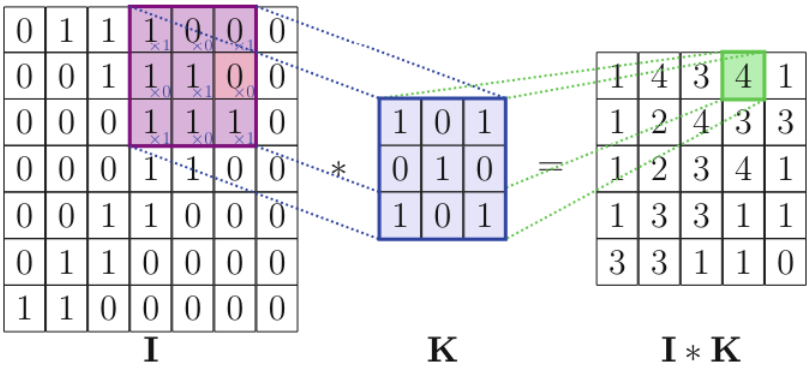


Figure 2.8: Convolution operation [8]

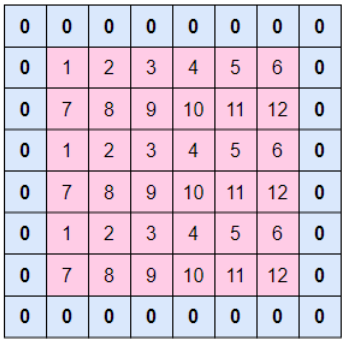


Figure 2.9: Zero padding for input image with P=1

2.4.2.2 Non Linearity Layer:

In this layer, a non-linear function f is applied to the resulting feature maps from the convolution layer as represented $\boldsymbol{r^k = f(h^k)}$ commonly in CNNs the non linear function is RELU (rectified linear unit)

2.4.2.3 Pooling layer:

Pooling or sub-sampling after nonlinear layer is for shrinking the size of feature maps by sliding kernel of size $n \times n$ over the feature map and replacing the $n \times n$ region in feature map with the maximum value or the average of region pixels hence there is two type of pooling: Max pooling, average pooling.

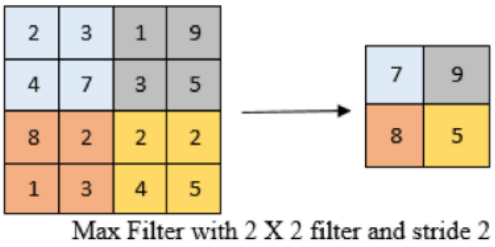


Figure 2.10: Max Pooling [9]

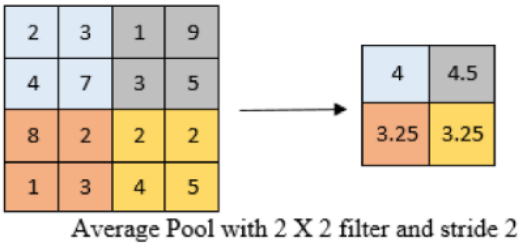


Figure 2.11: Average Pooling [9]

2.4.2.4 Fully Connected Layer:

It is the final step in CNN architecture, it is simply a feed forward neural network that receives an flatten input from the last pooling layer. In the context of image classification the last layer has a size equal to the number of classes (denoted as C) that the CNN is trained to classify.A softmax activation function is applied to the outcomes of feed forward neural network producing normalized probability scores. The class with the highest probability is considered the final output of the CNN model.[9]

2.4.3 Vision Transformers

Recently Vision Transformers become a popular alternative to Convolutional Neural Networks(CNNs) by displaying remarkable capacities in Computer vision tasks.[62]

Vision Transformers, also known as ViTs, are inspired by the self-attention mechanism from the Transformers architecture in NLP. Self-attention allows Transformers to learn the relationships between elements of a sequence, which results in attending to all elements for learning long-range relationships, compared with recurrent neural networks(RNNs) that deal with sequence elements recursively and only attend to short-term context.[63]

ViTs treat images as sequences of patches. First, it divides the input image into sequences of patches(Image patches) originally with a fixed size of (16x16) acts in the same way as tokens (words) in NLP application, then are transformed into embedding vectors by using linear projection. Position and patch embeddings are added together and then fed through a series of Transformers blocks. The ViT architecture is composed of tokenization, position embeddings, and the transformer encoder that contains a set of layers of multi-head self-attention react as one block completed with multi-layer perceptron block, Layer normalization preceding every block, and residual connections after every other block.[10] The figure 2.12 represents the key components of ViT and below it, we furthermore mention details about each component

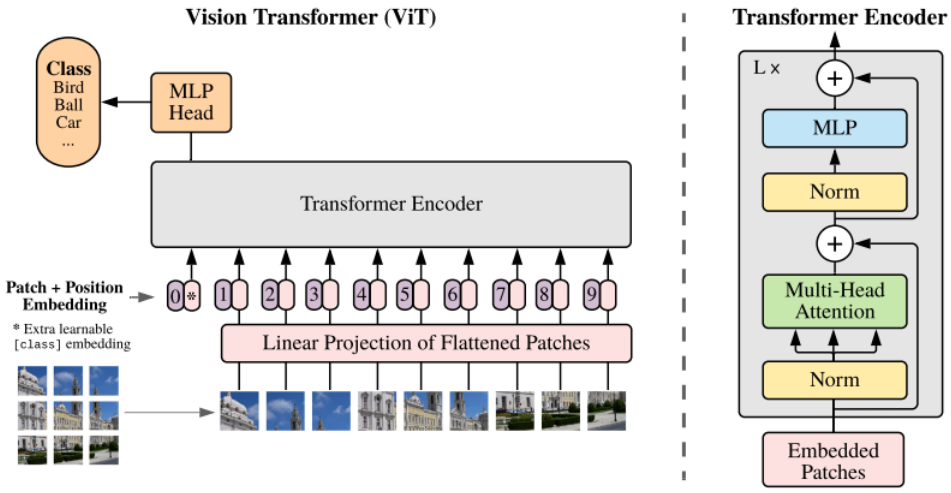


Figure 2.12: vit architecture and its components [10]

2.4.3.1 Toknization

Tokenization is an important step in preparing the input image for ViT, conventionally an input image $\boldsymbol{X} \in \mathrm{R}^{\boldsymbol{H}\times\boldsymbol{W}\times\boldsymbol{C}}$ and its following labels $\boldsymbol{Y}$, $\boldsymbol{X}$ gets reshaped to a sequence of flattened 2D image patches $\boldsymbol{X_p} \in \mathrm{R}^{\boldsymbol{N}\times(\boldsymbol{P^2}\cdot\boldsymbol{C})}$ where *(H,W)* is the height and width of input image, C is the number of channels, P denotes the patch size with N represent the number of patches. each patch is flattened and then passed to a linear layer, for mapping to D dimensions by multiplying a learnable weight matrix by every flattened patch embedding, the result is embeddings in lower-dimensional feature space because the Transformer has constant latent vector size D through all of its layers.

The *class* token is pre-append to the token sequence, it acts as a learnable parameter by the ViT model which attends to the most important features or regions of the image. the *class* token after inference of multiple transformer encoder blocks it feds to MLP for final classification output.[10][64]

2.4.3.2 Position Embedding

In order to save the relative or absolute positional information about patch embeddings (sequence tokens), learnable or pre-defined position embeddings that have the same d dimension as the model, are added to patch embeddings [65][66]. For encoding the positional information an adopted sine and cosine function with different frequencies:

$$\boldsymbol{PE_{(pos,2_i)} = sin(pos/100000^{2i/d_{model}})} \tag{2.4.10}$$

$$\boldsymbol{PE_{(pos,2_i+1)} = cos(pos/100000^{2i/d_{model}})} \tag{2.4.11}$$

Where *pos* is the position of the token of the sequence and i represents the index of the dimension in the embedding vector[65]. The combined patch and position embeddings are formulated

as follows:

$$\boldsymbol{z_0 = [x_{cls}; x_p{}^1 \cdot E; ...; x_p^N \cdot E] + [E_{pos}^{cls}; E_{pos}^1; ...; E_{pos}^N]} \tag{2.4.12}$$

where $\boldsymbol{x_{cls} \in R^D}$ is the class token,$\boldsymbol{E \in R^{N \times (P^2) \times D}}$is a linear projection of each patch $\boldsymbol{X_p}$, and $\boldsymbol{E_{pos}^i \in R^D}$is the position embedding for the i-th token, then the newly formulated inputs are fed to the transformer blocks [10].

2.4.3.3 Self-Attention (Attention)

The foundational mechanism in the Transformer architecture is known as "Scaled Dot-Product Attention," a concept introduced by Vaswani et al. (2017) [65]. The inputs consist of a query vector and set of key-value vector pairs, it measures the similarity score between each query with the key vectors by using the dot product operation. The equation is expressed as:

$$\boldsymbol{Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V}$$

Where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ with $\boldsymbol{d_k}$, denote the dimensionality of the queries and keys, respectively, with the values being of dimension $\boldsymbol{d_v}$. key, query, and value are the transformations of the initial patch embeddings using linear projection with a set of learnable weights matrices $\boldsymbol{W^K}$, $\boldsymbol{W^Q}$, and $\boldsymbol{W^V}$ respectively, the learnable weights matrices are randomly initialized at the beginning of the training step.

The dot-product between the query and key pair results in a set of similarity scores then are scaled by $\frac{1}{\sqrt{d_k}}$ as a scaling factor to prevent the saturation of softmax [67]. The scaled scores are passed to Softmax to obtain the attention scores. A weighted sum of the value vectors with attention scores acting as weights is the self-attention's final result.

2.4.3.4 Multi-Head Attention

As mentioned before in the attention mechanism query, key, and value are the result of linear projection using learnable weight matrices. Furthermore, this linear projection is performed h times with h represents the number of heads in the MultiHead attention function, where each head has its set of learnable $\boldsymbol{W^K}$, $\boldsymbol{W^Q}$, $\boldsymbol{W^V}$ weights matrices and it performs attention to the newly transformed representations of the input patch embeddings with all attention heads work in parallel for efficient information process and capturing different aspects of the input information[67]. The matrices resulting from the parallel processing (MHSA output) of attention heads are concatenated and then linearly transformed by another learnable matrix, denoted as $\boldsymbol{W^O}$ Formally:

$$\boldsymbol{MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O,}$$

$$\boldsymbol{head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)}$$

$\boldsymbol{W_i^Q \in R^{d_{model} \times d_k}}$, $\boldsymbol{W_i^K \in R^{d_{model} \times d_k}}$, $\boldsymbol{W_i^V \in R^{d_{model} \times d_k}}$ and $\boldsymbol{W_i^O \in R^{hd_v \times d_{model}}}$ are the learnable matrices used to project the inputs and output vectors in the attention operation respectively.

2.4.3.5 Feed Forward Network (FFN)

A feed-forward network, known as Multi-Layer Perceptron, is utilized to process the outcomes of self-attention computation on the input data. This fully connected layers has one hidden layer. Formally, given an input x, and learnable weights and biases W1,b1, W2, b2 and an activation function $\boldsymbol{\rho}$ [67], the MLP can be expressed as:

$$\boldsymbol{MLP(x)} \;=\; \boldsymbol{\rho(xW1 + b1)W2 + b2} \tag{2.4.13}$$

2.4.3.6 Layer Normalization

Before and After every multi-head attention a layer normalization is applied to the embeddings [66] as follows:

$$\boldsymbol{LN(x)} = \frac{\boldsymbol{x - \mu}}{\boldsymbol{\sigma}} \odot \gamma + \boldsymbol{\beta} \tag{2.4.14}$$

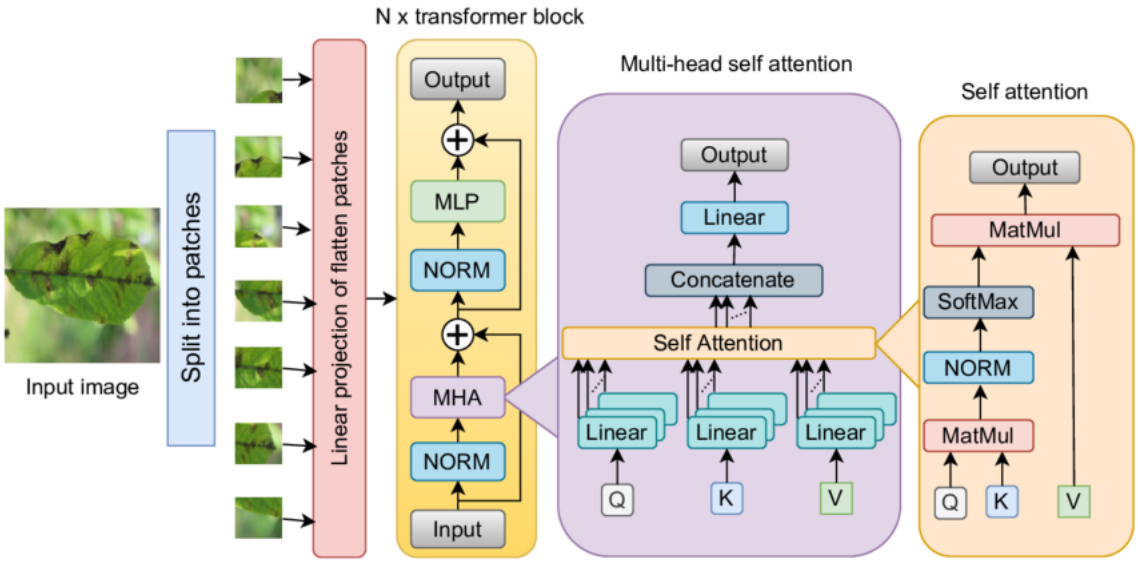


Figure 2.13: Overall ViT workflow [11]

2.4.4 Pre-Trained Models

Deep learning models particularly in the computer vision domain, typically require a large dataset to learn meaningful and efficient representations that can be leveraged for achieving good performance in various tasks. however in some real-world scenarios, the provided data is limited, and collecting more data may be costly or impractical. For example, using crowd-sourcing to segment images costs about $6.4 per image, and some medical imagining tasks require annotation from an expert leading to a cost of much more to build a dataset [68].

Pre-trained models are models that were trained on large-scale datasets(e.g. ImageNet) for a certain task and they can be reused or fine-tuned on different tasks by transfer learning, making solving new task effectively and fast, while also requiring significantly less data and computational resources compared to training a model from scratch.

Below are the most popular pre-Trained models:

2.4.4.1 Visual Geometry Group (VGG):

VGG is a classical convolutional neural network (CNN) architecture, developed to increase the depth of such networks by utilizing small 3 x 3 filters and consisting of pooling layers and a fully connected layer. it was first introduced by two Oxford researchers at the visual geometry group lab in 2014 and has been influential in the field of computer vision because it demonstrated excellent performance on various image classification and recognition benchmarks including the ImagNet large-scale visual recognition challenge(ILSVRC). VGG16 and VGG19 are two models of the VGG architecture, with 16 and 19 convolutional layers, respectively [12].

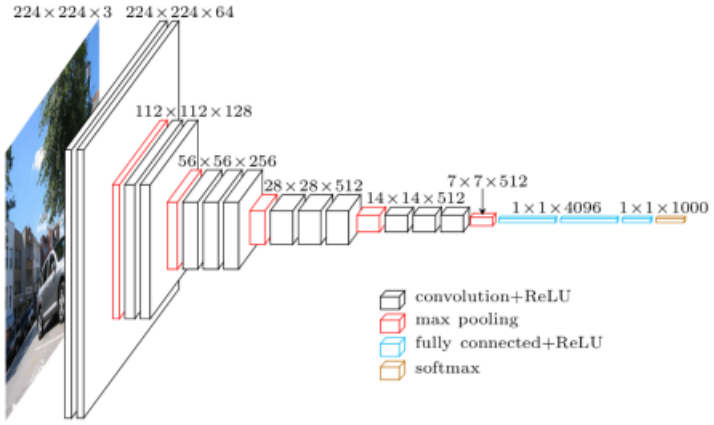


Figure 2.14: The structure of the VGGNet model [12]

2.4.4.2 Residual Neural Network (ResNet):

ResNet is a significant deep learning model where weight layers learn residual functions with respect to the layer inputs. ResNet was developed in 2015 for image recognition and excelled in the ImageNet challenge. It is based on different types of residual blocks, including basic blocks consist two 3x3 convolutional layers with a residual connection, bottleneck blocks that have three sequential convolutional layers for dimension reduction and restoration, and pre-activation blocks to reduce non-identity mappings between blocks.It address the vanishing/exploding gradient problem by utilizing identity skip connections that allowing the gradient flow along an extra path (shortcut path). There are different variants of the ResNet such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 [69].

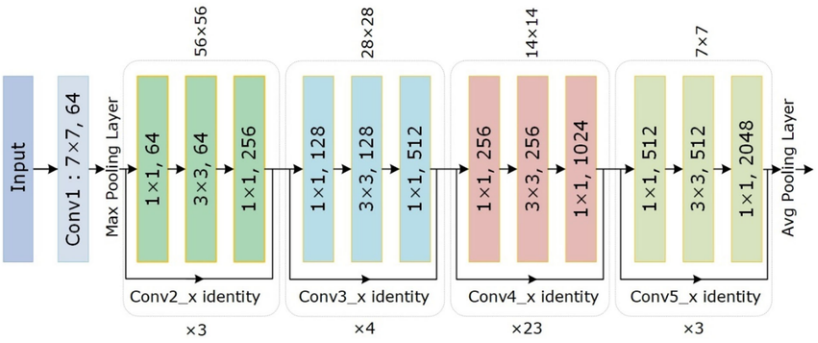


Figure 2.15: The general architecture of ResNet [13]

2.4.4.3 MobileNet:

MobileNet is a type of convolutional neural network known for its lightweight design for mobile and embedded vision applications. It is based on a streamlined architecture that uses depthwise separable convolutions rather than standard convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices [70].

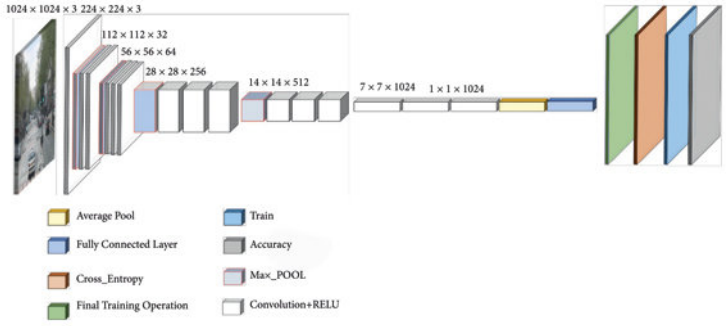


Figure 2.16: The structure of MobileNet V1 [14]

2.4.4.4 SqueezeNet:

SqueezeNet is a convolutional neural network that was developed to address the advantages of smaller DNN architectures, such as reduced communication during distributed training, less bandwidth for model export, and feasibility for deployment on hardware with limited memory, by the fire module which is the foundation of SqueezeNet that designed according to replace 3 × 3 filters with 1 × 1 filters, reduce the number of inputs for the remaining 3 × 3 filters and late downsampling in the network so that convolution layers have large activation maps, to offer high performance with smaller model size compared to traditional networks [71].

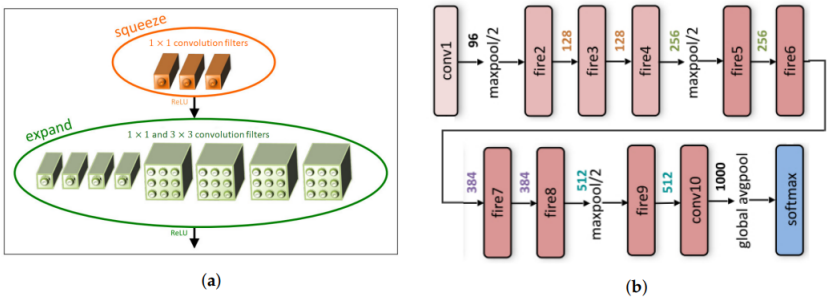


Figure 2.17: (a) Fire module in SqueezeNet (b) SqueezeNet architecture [15]

2.5 Image Segmentation

Image segmentation is a fundamental task in computer vision that involves dividing an image (or video frames) into multiple meaningful regions to simplify its representation and facilitate its analysis. Image segmentation goes beyond simple classification of the entire image and delves

into understanding the content. Segmentation algorithms segment images or regions with specific characteristics using low-level image concepts such as pixel density, pixel color, and texture [51].

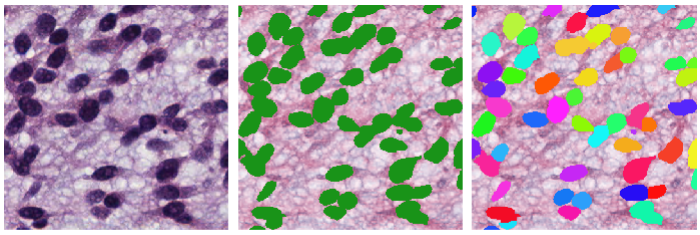
Image segmentation can be formulated as a classification problem of pixels with semantic labels (semantic segmentation) or segmentation of individual objects (instance segmentation). Semantic segmentation performs a classification of all image pixels with a set of object classes (e.g., human, car, tree, sky), while instance segmentation detects and identifies every object of interest in the image with differentiates between different instances of the same object (e.g., partitioning of individual persons).

Segmentation is essential in various applications such as medical image analysis (e.g., tumor boundary extraction and measurement of tissue volumes), autonomous vehicles (e.g., pedestrian detection, road boundaries, and other vehicles), robotics, video surveillance, and augmented reality [72].

2.5.1 Image segmentation for medical image analysis

Medical image segmentation is one of the basic techniques and challenging tasks in medical image analysis. It refers to identifying pixels of an organ or lesion (region of interest) from medical images and extracting crucial information about their shapes and sizes.

Traditional methods are mainly based on handcrafted designs features extracted by a domain expert based on image processing and mathematical techniques such as thresholding [73, 74], edge detection [75, 76], and morphological operations [77]. These methods are based on arbitrary parameters and often suffer from accuracy and adaptability due to the complexity and diversity of medical images and the inevitable manual intervention in it, which makes it extremely difficult to implement especially when dealing with a large number of instances [78].



(a) Original image (b) Semantic seg. (c) Instance seg.

Figure 2.18: Different types of segmentation [16]

Deep learning methods, that based on convolutions neural networks (CNNs) [79, 17, 18] and Transformers [19, 20, 21], have shown superior accuracy, efficiency, and automation in medical image segmentation tasks. below are the most popular models that are widely used in medical image segmentation:

2.5.1.1 Image Segmentation Based on CNNs

- **U-Net:** The U-Net is a convolutions neural network architecture commonly used for medical image segmentation tasks. It was first introduced in 2015 by Ronneberger et al for the ISBI Challenge. and was able to profit from the characteristics of convolutions neural networks and fully convolutions networks, in addition to benefiting from low and high-level features via skip connections [17]. The U-Net architecture consists of a:

Contracting Path (Encoder) captures context by reducing the spatial dimensions and increasing the number of feature channels using a series of convolutions and pooling layers.

Symmetric Path (Decoder) to precise localization and allows the network to retain spatial information lost during the encoding using up-sampling and concatenating them with the corresponding feature maps from the contracting path.

Skip Connections to connect the contracting path to the expanding path that helps preserve fine-grained details and gradients during training and makes the segmentation more accurate

To create the segmentation mask, the final convolution layer uses a pixel-wise classification method to categorize each pixel in the input image into a certain class. There are many subsequent variations of u-net like UNet++ [80], Attention U-Net [81], and Deep Residual U-Net [82] that were developed to enhance its performance in specific tasks.

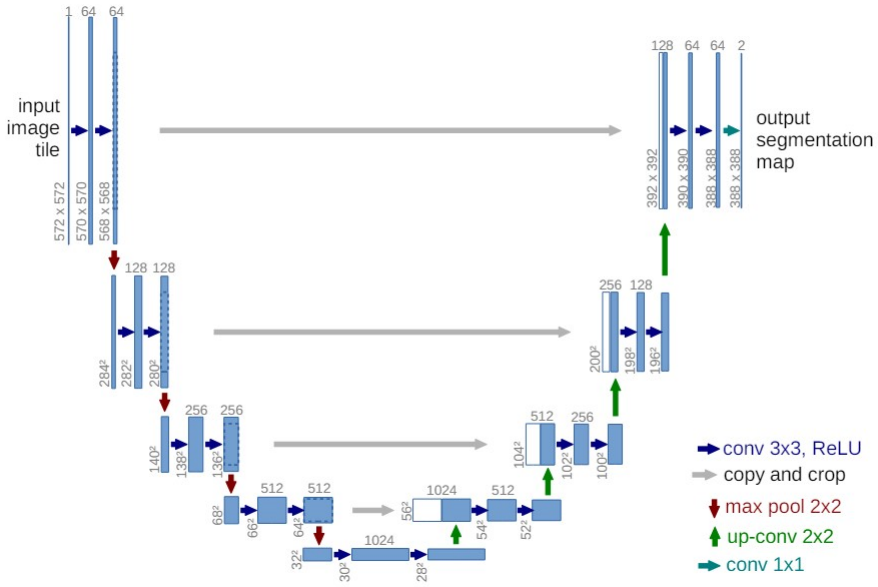


Figure 2.19: The architecture of U-Net[17]

- **SegNet:** SegNet is a deep convolution encoder-decoder architecture designed for image segmentation tasks. that focuses on memory efficiency and computational speed. It consists of an encoder network and a corresponding decoder network, and a pixel-wise classification layer. The encoder network is similar in structure to the VGG16 network, whereas the decoder network uses pooling indices from the encoder to perform non-linear up-sampling, eliminating the need to learn to up-sample. This design choice makes SegNet efficient in terms of memory consumption and computational time during inference [18].

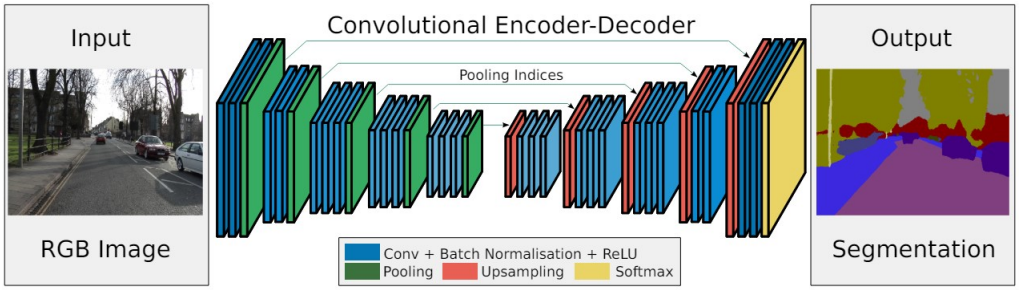


Figure 2.20: The architecture of SegNet [18]

2.5.1.2 Image Segmentation Based on Vision Transformers

- **TransUnet:** TransUNet is a hybrid model that combines Transformer and U-Net architectures, It enhances the performance of different image segmentation tasks by utilizing local U-Net information in addition to the global self-attention processes of Transformers. It is composed of a Transformer encoder that extracts global contexts by tokenizing image patches from a convolution neural network (CNN) feature map. This self-attentive feature encoded is then up-sampled to be combined with various high-resolution CNN features that were skipped from the encoding path, to enable precise localization [19].

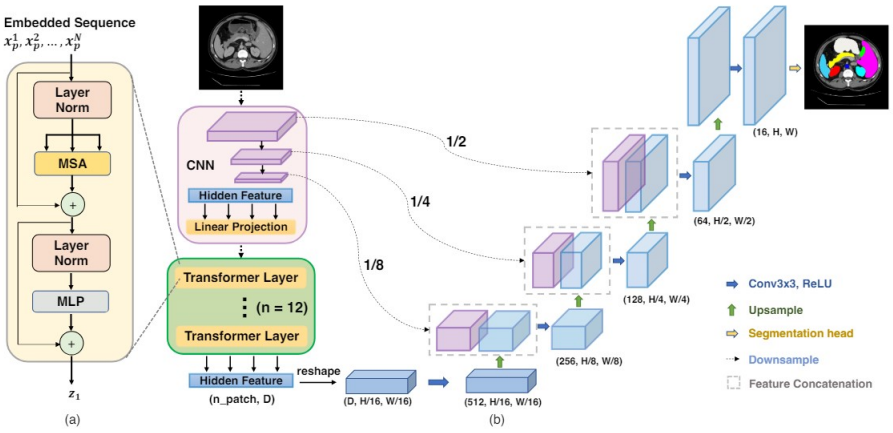


Figure 2.21: The architecture of TransUnet[19]

- **Swin-Unet:** Swin-Unet is a Transformer-based U-shaped Encoder-Decoder architecture with skip connections designed for medical image segmentation. It makes long-range and global semantic information interaction easier. The encoder utilizes a hierarchical Swin Transformer with shifted windows to extract context information, while the decoder employs a symmetric Swin Transformer-based design with a patch expanding layer for up-sampling to restore the spatial resolution of the feature maps [20].

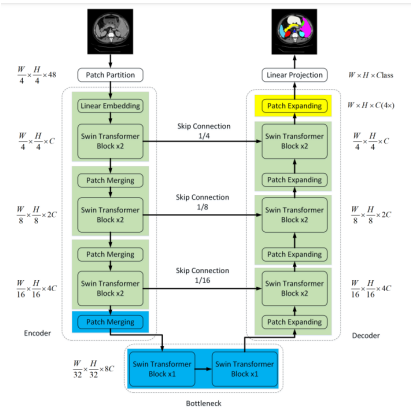


Figure 2.22: The architecture of Swin-Unet [20]

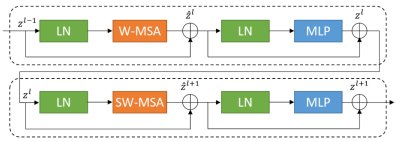


Figure 2.23: Swin transformer block that compose LayerNorm (LN), window and shifted window-based multi-head self attention module (W-MSA and SW-MSA), residual connection and 2-layer MLP with GELU non-linearity. [20]

- **U-Transformer:** U-Transformer combines a U-shaped architecture with attention mechanisms from Transformers, allowing it to model long-range contextual interactions and spatial dependencies crucial for accurate segmentation in challenging contexts. Attention mechanisms are incorporated at two main levels: a multi-head self-attention module uses global interactions between semantic features at the encoder to explicitly model full contextual information while multi-head cross-attention in the skip connections filter out non-semantic features, allowing a fine spatial recovery in the U-Net decoder. Many medical image segmentation applications have demonstrated notable performance with this architecture [21].

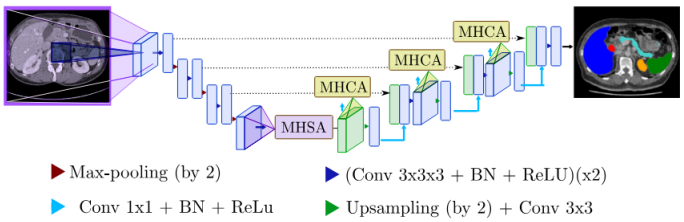


Figure 2.24: The architecture of U-Transformer [21]

Although these models have achieved remarkable success in image segmentation, they suffer from some limitations include computationally expensive and requires retraining for new tasks [83], focuses primarily on local features and struggle to effectively fuse low-level and high-level information that can lead to difficulties in segmenting objects with complex shapes, intricate details, or occlusions [84], Struggle with generalizing to unseen variations in object appearance, lighting conditions, or image backgrounds that can lead to inaccurate segmentation results [85], and heavily rely on large labeled datasets for training and the quality of the images that can be expensive and time-consuming to acquire, especially for specialized applications [86].

2.6 Segment Anything Model (SAM)

The recent rapidity towards general artificial intelligence by developing an AI system that generalizes to a wide range of tasks while demonstrating an intelligence level comparable to that of a human being. As a result, an evolving concept named foundation models represents an AI system that is pre-trained on web-scale datasets and has the capability of zero-shot generalization on a wide range of tasks [22].

Large Language models e.g. GPT-3[87], T5[88], and GPT-4[89] are considered foundation models that proved a great performance in handling a variety of Natural Language Processing (NLP) tasks such as Text-summarization, Question answering, information retrieval, and Machine translation. Consequently, the Computer Vision community started exploring large visual models (LVMs) by scaling vision transformers to huge sizes via pre-training on large datasets and the techniques of incorporating knowledge of different modalities (Text, audio) into LVMs [22].

In April 2023, Meta AI launched a new segmentation model named "Segment Anything model" [26], it is an AI model that can "cut out" any object in any image using a prompt that specifies what to segment. Segment Anything model is considered a large vision model made for general image segmentation that was pre-trained on 11 million images with 1 billion masks.

The extensive training of SAM allowed it to learn a general notion of what objects are, this key competence enables a zero-shot generalization to unfamiliar objects and images without additional retraining by leveraging prompts similar to an LLM. [22].

SAM architecture is composed of 3 parts Image Encoder, Prompt Encoder, and Image decoder illustrated in figure 2.25.

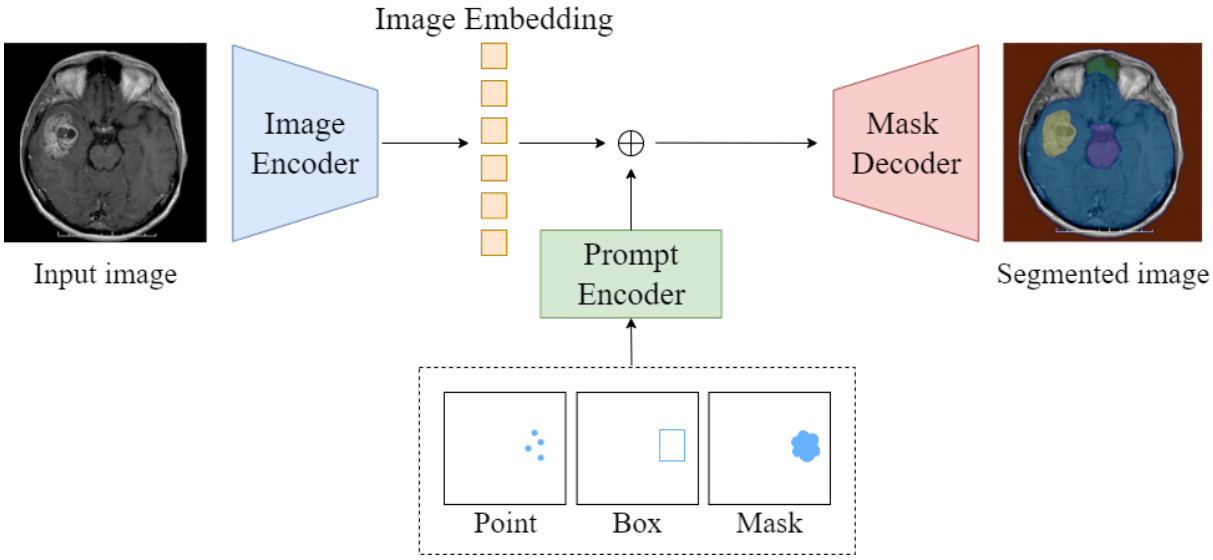


Figure 2.25: Overveiw of SAM architecture[22]

Segment anything model generates masks for a given input image by applying an image encoder based on vision transformers (ViT) to extract rich image embeddings. The encoder comes in three sizes: ViT-B (Base), ViT-L (Large), and ViT-H (Huge), with different parameter scales 91M, 308M, and 636M respectively. Mask, point, box, or text provided by user interactions are encoded using a prompt encoder then mask decoder incorporates the information from these prompts embeddings along with the image embedding to predict a valid mask.

2.7 Related Works

In this section, we establish the context and relevance of the current works on applying SAM to medical image segmentation (MIS) with specific attention to the Polyp Segmentation task, the provided comprehensive overview includes the essential current state of knowledge about SAM and identifying the gaps that this research aims to address. Critically examining the theories, methodologies, and findings of previously published studies.

To have a better understanding of the recent findings, we suggest categorizing the related works based on the architecture of SAM models: works on the encoder, works on the prompts, and works based on the decoder are outside the scope of this work. We focused our attention on the first two categories that related to our work. This categorization will provide insights into the different architectural components of SAM and how they have been explored to adapt it for MIS tasks like polyp segmentation.

2.7.1 Designing effective prompts:

A recent comprehensive study by [90] systematically evaluated the performance of SAM using three auto prompt modes, box prompts, and point prompts illustrated in figure 2.26, on 12 different public medical image datasets covering CT, X-ray, MRI, Endoscopy, Ultrasound, and OCT, for finding the suitable prompt mode for medical image segmentation using SAM. The conducted experiments revealed that overall the box-prompt achieves the highest segmentation accuracy among the three modes. it also observed that when positional jitters(noise added to the bounding box coordinates) ranging from 0.01 to 0.5 are added to the bounding box resulting in a drop in overall segmentation performance on 12 datasets specifically on the Polyps segmentation results drop down from 0.9086 to 0.5438 in terms of Dice score. This indicates that SAM is sensitive to the quality of the provided bounding box prompt. In addition, SAM's zero-shot performance was generally lower than full-supervised SOTA models but when an effective prompt (bounding box generated from ground truth mask) is given it achieved competitive or exceeding results on several datasets.

Figure 2.26: Three prompt variations used : a) The auto prompt SAM will be automatically prompted with a regular grid of points and predicate a set of masks for each point prompt then select the high-quality masks with non-maximal suppression. c) Bounding box prompt generated from the ground truth mask. b) The center of the bounding box is chosen to be as point prompt.

In [91] they introduced few-shot medical image segmentation by leveraging the Segment Anything model (SAM). The key idea is to utilize a simple linear pixel-wise classifier to self-prompt SAM, a prompting unit takes the image embeddings from SAM's encoder, and a Logistic regression model is trained to predict low-resolution masks (coarse masks) using the ground truth labels. The coarse masks provide information about the location and size of the segmentation target, where the bounding box represents the location extracted from the coarse mask and the distance transform of the coarse mask to find one point that represents the location.The authors evaluate their method on the Kvasir-SEG and ISIC-2018 Polyp datasets by training the logistic regression model on a few training images (e.g., 20 images). Then extracting the prompts for all images, the proposed method results 62.78%Dice, 53.36%IoU on Kvasir-SEG and 66.78%Dice 55.32%IoU on ISIC-2018 surpassing other fine-tuning methods like MedSAM and SAMed with only 20 training images.

The research [92] tilted 'Polyp-SAM++: Can A Text Guided SAM Perform Better for Polyp Segmentation?' focuses on prompting with text prompt for guiding SAM on the polyp segmentation task. The text prompt creates a bounding box using GroundingDINO. [93] which is a zero-shot approach.The key results and contribution related to few-shot ability are achieving better performance compared to existing CNN, Transformer, and SAM-based models for polyps segmentation on three benchmark datasets (Kvasir-SEG, CVC-300, CVC-ClinicDB) . For example, on the kvasir-seg dataset, Polyp-SAM++ achieved a mean dice of 0.90%, mIoU of 0.86, outperforming SAM-H(0.77), and SAM-L(0.78). This simple approach of incorporating text prompts with SAM improves overall polyp segmentation performance compared to using an unprompted SAM.

However, in [90], the study did not explore SAM's fine-tuning capabilities on selected datasets with limited labeled data, which could potentially bridge the gap between the zero-shot performance and the performance of supervised models. They also highlight the importance of designing appropriate prompts that are not based on a ground truth mask (zero-shot setting) while exploring other automated prompt generation strategies that could further improve SAM. Additionally, the use of a simple linear classifier (Logistic Regression) to generate prompts in [91] limits the quality of generated prompts, which further limits SAM's performance. Engineering prompts that describe the desired segmentation in a human-like supervision manner is crucial for example prompts that describe the shape, color, size, and location of brain cancer in MRI scan.

2.7.2 Strategies of adapting the encoder of SAM to the target domains (MIS):

In [94] a novel approach is proposed for fine-tuning SAM for medical image segmentation called Ladder Fine-tuning (LST) . It combines a CNN encoder with SAM architecture with integrating a learnable gate that combines features from SAM encoder and the new CNN encoder. Fine-tuning part is only for the CNN encoder and SAM decoder while keeping the large SAM encoder frozen, significantly reducing the training time and computational resources compared to adapter and other fine-tuning methods that update the full SAM model. This research addresses the challenge of adapting SAM to medical domain using an fine-tunning strategy that is cost friendly and effective. They used synapse multi-organ segmentation dataset of the MICCAI 2015 Multi-Atlas abdomen labeling challenge. It includes 30 abdominal CT scans that each CT scan has 8 abdominal organ. 18 scans used for training and 12 scans are used for test the reported results are in Dice Similarity Coefficient (DSC) and Hausdorff Distance (HD95).The

LST accomplished results of 79.45% DSC and 35.35mm HD95 surpassing SOTA methods.

In This study [23] a new fine-tunning strategy that employs LoRA (Low Rank adaption) for customizing SAM on MIS namly SAMed. the fine tuning strategy adding small subset of trainable parameters (low-rank matrices/ LoRA layer) inside each transformer block in the SAM encoder, where LoRA layer acts as skip connection before and after projection of queries and projection of values of the transformer block.then finetuning the SAM encoder together with the prompt encoder and the mask decoder the figrue illustrate the full framework of SAMed. LoRA allows SAM to update a small fraction of parameters during training on medical image. They adopted synapse multi-organ segmentation dataset of MICAAI 2015 for evaluation, the obtained results 81.88% DSC and 20.64HD.

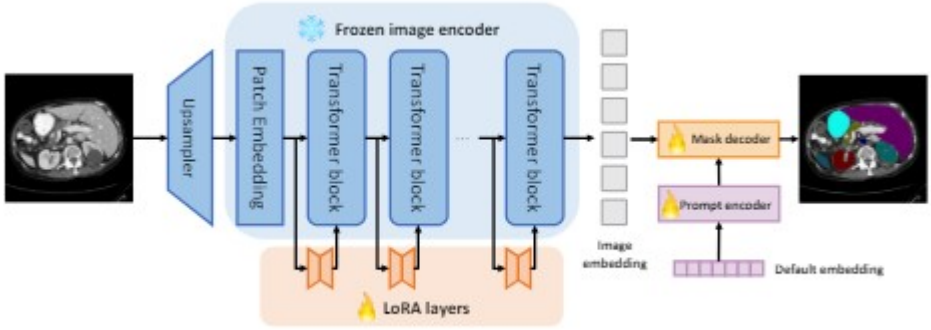


Figure 2.27: The framework of SAMed [23].

A transfer learning strategy for fine-tuning SAM to polyp segmentation was implemented in [95] with the name Polyp-SAM. They explore two transfer learning strategies: (a) fine-tuning only the mask decoder while freezing the encoders, and (b) fine-tuning all components of SAM, including the image encoder, prompt encoder, and mask decoder. Five public datasets CVC-ColonDB, CVC-300, Kvasir, and CVC-ClinicDB have been selected for evaluation, Polyp-SAM achieves performance on all datasets with a minimum of 88% Dice scores. The results demonstrate the potential of adapting SAM to medical image segmentation tasks.

although fine-tuning all components of SAM as in [95] yields satisfactory results but still this adaptation strategy is a computationally expensive technique, and the SAMed strategy faces overfitting when trained on limited data as shown in figure 2.28. The experiment was conducted in [24] to evaluate different adaptation strategies in a data-scarce setting.

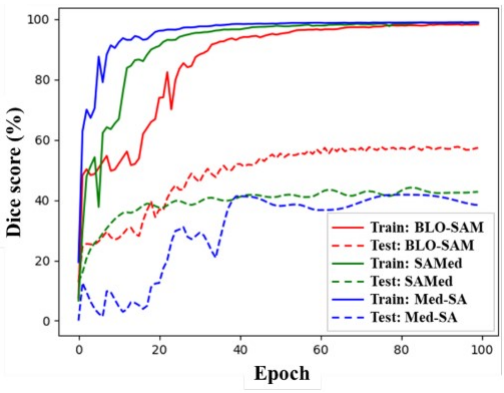


Figure 2.28: Dice scores on the train and test sets on gastrointestinal disease in three adaptation strategies when trained to a limited number of data BLO-SAM[24], SAMed[23] and MedSAM[25].

2.7.3 Summary

It is evident that in the current approaches for adapting SAM to medical image segmentation, a low-cost efficient parameter fine-tuning strategy should be employed along with generating prompts that further enhance SAM, resulting in high-quality segmentation masks.

Chapter 3

Proposed Method

3.1 Introduction

In this chapter, we introduce a deep learning model for polyp segmentation with zero-shot prompting that is based on the SAM architecture. The architecture aims to adapt SAM for medical segmentation, we first focused on reducing SAM's computational complexity by eliminating the SAM's huge ViT encoder as the opposite of the paper ladder fine tuning strategy [94] and we propose CNN encoder as

3.2 The Proposed Architecture

3.2.1 Prompt Encoder

The prompt encoder is essential for SAM's image segmentation, utilizing different input prompts such as points, bounding boxes, and text descriptions [1] to guide the segmentation process and direct the model's attention to particular areas or objects within an image [96].

The prompt encoder converts the various types of user-provided prompts into embeddings. This conversion process ensures that the prompts are represented in a format that the model can effectively utilize during segmentation. SAM can effectively interpret and combine different prompt types with the image features, enabling its powerful zero-shot generalization capabilities, and can handle two main types of prompts:

3.2.1.1 Sparse Prompts

These include points and bounding boxes, they allow the model to understand the precise areas of interest specified by the user.

- **The point prompt** represents the point's location on the input image with the label that indicates if the point is either foreground or background, it is encoded by the sum of the positional encoding [97] using random spatial frequencies, and the learned embeddings that dictate the point label either foreground or background [26].
- **The bounding box prompt** representation is an embedding pair, the first pair stands for the 'top-left corner point' and the second pair is the 'bottom-right corner point', where each

[1] The official implementation of text prompts in SAM is not yet provided by the authors in the codebase.

pair embedding is the sum of positional encoding of the point with a learned embedding that indicates it is either 'top-left' or 'bottom-right' [26].

3.2.1.2 Dense Prompts

- The **Mask** prompt get downscaled to 4x lower resolution than the input images then applying two 2x2,stride-2 convolutions with output channels 4 and 16, respectively. With a final 1x1 convolution that maps these channels to 256 channels. After each convolutional layer is further improved with GELU activations and layer normalization. The resulting mask embedding is added element-wise to the image embedding. If there is no mask prompt, a learned embedding express 'no mask' is added to each image embedding location. [26]

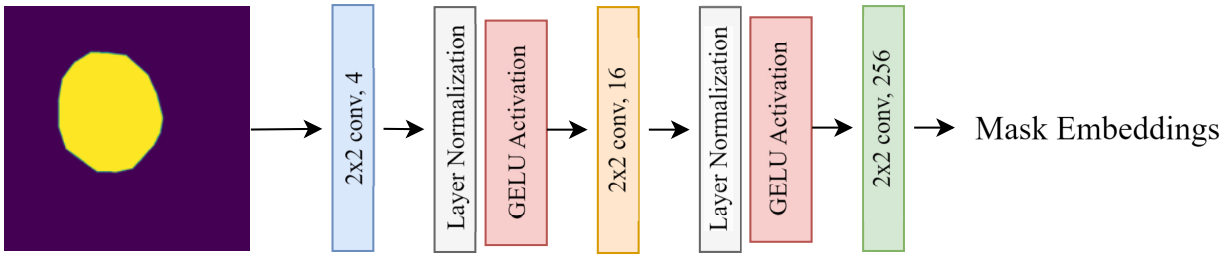


Figure 3.1: Mask Embedding Process

in the proposed architecture we used bounding box as prompts over point and mask prompts due several factors:

1. In medical applications requiring pixel-level mask annotations is labor-intensive and time-consuming. In contrast bounding box annotations around desired regions is much easier and faster to obtain .
2. Possibility of integrating in clinical workflows, where radiologists draw bounding box around suspected abnormalities during review.
3. The placement of point prompts requires some care to sufficiently capture the object.
4. Bounding boxes are less sensitive to minor annotation inaccuracies compared to mask prompts.

3.2.2 Mask Decoder

The mask decoder is responsible for generating a valid output mask by merging the image embeddings with the prompt embeddings. It employs two types of attention blocks, one for the prompt embedding and the other for the image embedding. These attention blocks allow the mask decoder to focus on the relevant parts of the image based on the provided prompt.

Prior to executing the decoder, a learned output token embedding is inserted into the set of prompt embeddings, resulting in new set of embeddings named 'Tokens'. It is noteworthy that the learned output token acts as a *class* token in ViT architecture [26, 10].

The decoder is composed of two layers as shown in figure 3.2, each layer performs 4 operations:

1. Self-attention on the tokens.
2. Cross-attention from tokens (as queries) to the image embedding.
3. A point-wise MLP updates each token.
4. Cross-attention from the image embedding (as queries) to tokens.

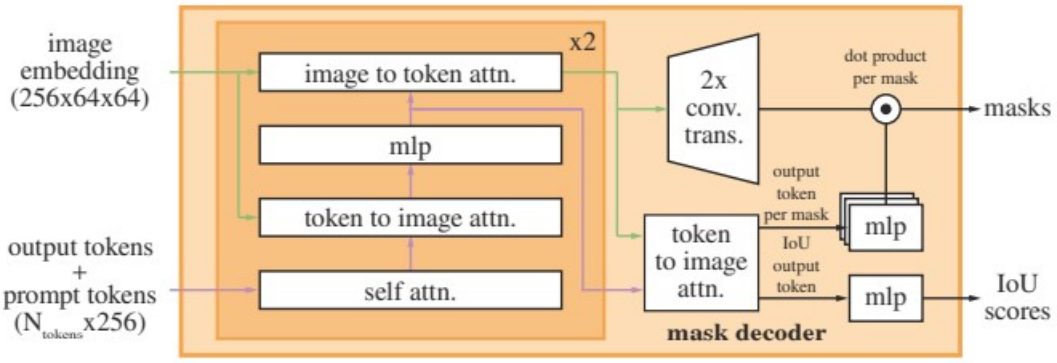


Figure 3.2: The lightweight mask decoder. A two-layer decoder updates both the image embedding and prompt tokens via cross-attention.[26]

The final stage of cross-attention in the decoder layer is for updating the image embeddings with prompt information. Each transformer self/cross-attention and MLP has a residual connection, layer normalization, and dropout of 0.1 during the training process. Then the first decoder layer transfers the updated tokens and image embeddings to the second decoder layer for the repetition of the four-step process[26].

After running the decoder, the updated image embeddings are upsampled with two transposed convolutional with 2x2 kernel size, stride 2, and GELU activations along with output dimensions of 64 and 32 consistently, they are separated by layer normalization. Then tokens attend to the image embeddings, the updated output token is passed to a 3-layer MLP that outputs a vector that matches the dimension of the up scaled image embedding. A dot product between the up scaled image embedding and the previous MLP's output for generating masks.

The output of the mask decoder is a set of binary segmentation masks that outline the objects or regions of interest (ROI) in the image and estimated quality score (IoU scores) for each generated mask, inducting the model's confidence in the accuracy of the segmentation [98].

Chapter 4

Experimental Results

4.1 The Experimental Dataset

For the evaluation of the Polyp segmentation task, we used a well-known dataset named Kvasir-SEG [99]. It contains 1000 gastrointestinal polyp images that were captured using colonoscopies. each image is accompanied by a segmentation mask illustrated in figure 4.1. Both Images and masks are encoded in JPEG format. The image resolutions range from 332x487 to 1920x1072 pixels. Additionally, a JSON file is attached that contains the bounding box (coordinates points) information about the polyp in the image. In the context of our research, the bounding box information is used during training and inference for guiding the attention mechanism and improving overall segmentation quality.

In addition, the authors of the dataset provide a data split[1] consisting of 880 images for the training phase and the remaining 120 images for validation/testing. We further split this set randomly into 60 images for validation and 60 images for testing, the architecture was trained on the provided split.

[1]the provided data-split https://github.com/DebeshJha/Kvasir-SEG/tree/main/Data-split

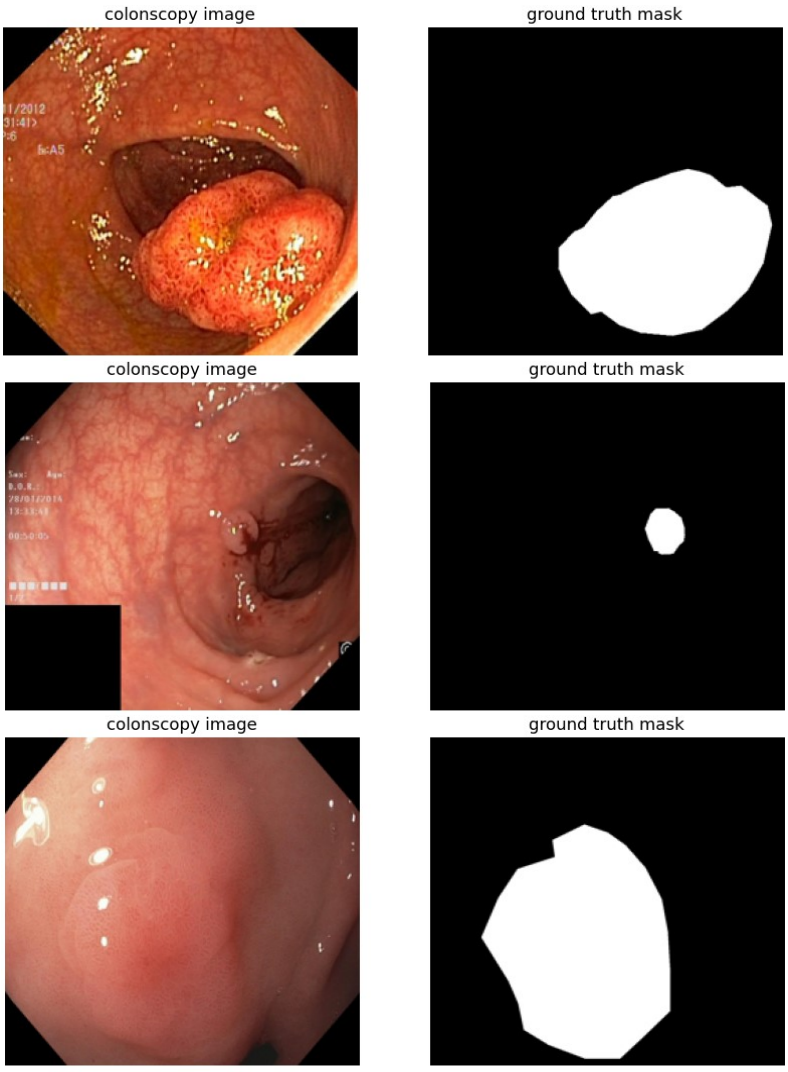


Figure 4.1: Samples images from the Kvasir-SEG dataset and their corresponding masks

4.1.1 Dataset Analysis

The task of accurately segmenting polyps from colonoscopy images is challenging because of several factors including:

- There are several polyp cases where the boundaries between the polyp and the colon tissue can be unclear, further complicating the segmentation task.
- The existence of multiple polyps in one single colonoscopy image that have different sizes is shown in figure 4.5, ranging from as small as 5mm to over 50mm in diameter. This variability in the number, size, and appearance of polyps within the same image adds another layer of complexity.

All these challenges impose an impact on the overall performance of any proposed polyp segmentation architecture, especially the data imbalance illustrated in figure 4.2. The figure 4.3

represents the distribution of the different sizes of polyps in relation to the image, while figure 4.4 shows some examples of the different polyps sizes.

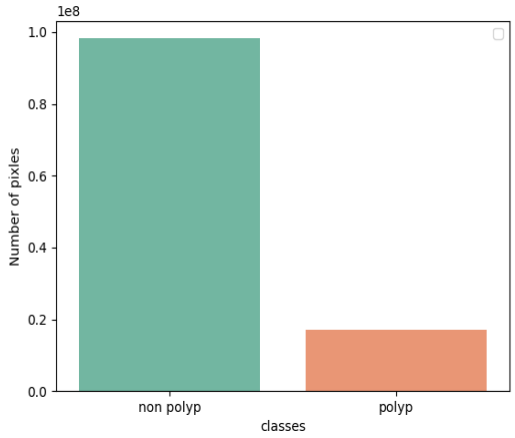


Figure 4.2: Distribution of the classes in the Kvasir-Seg dataset

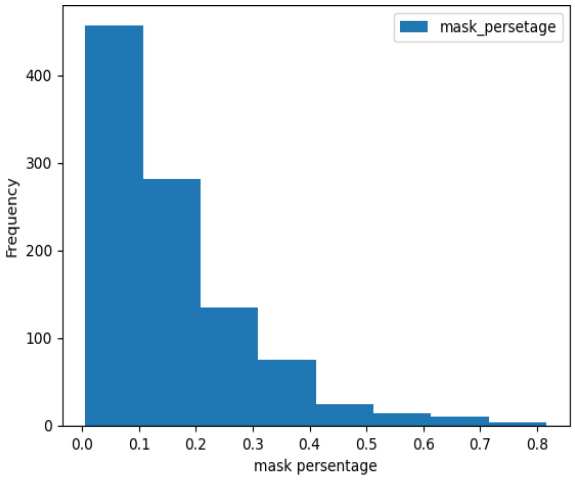


Figure 4.3: Frequency of images based on the relative size of the polyp in the image

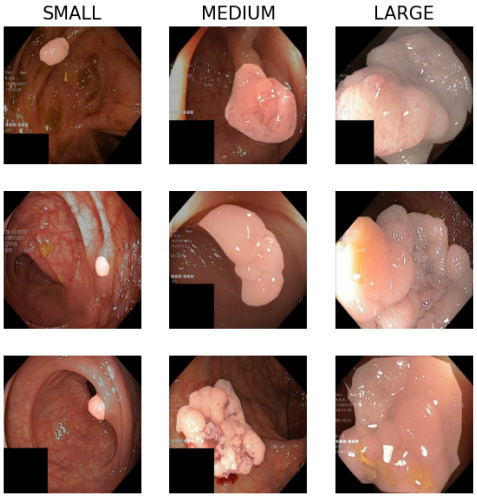


Figure 4.4: Different polyp size in the Kvasir-Seg dataset

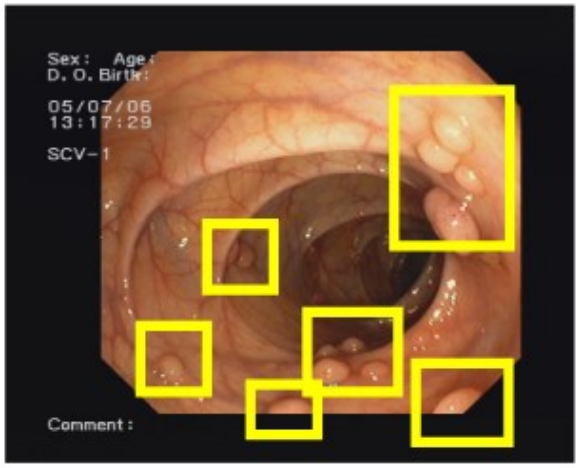


Figure 4.5: Multiple Polyps in one single image [27]

4.1.2 Data Pre-Processing

Data pre-processing is a crucial step in preparing a dataset to a format that the model can accept. By improving image quality, lowering noise, and standardizing the input data, it facilitates more accurate segmentation model performance.

Several pre-processing steps have been applied to the dataset:

- **Min-Max Scaling** Also known as normalization is a common technique in machine learning and statistics that scales numerical features to a specific range, typically between 0

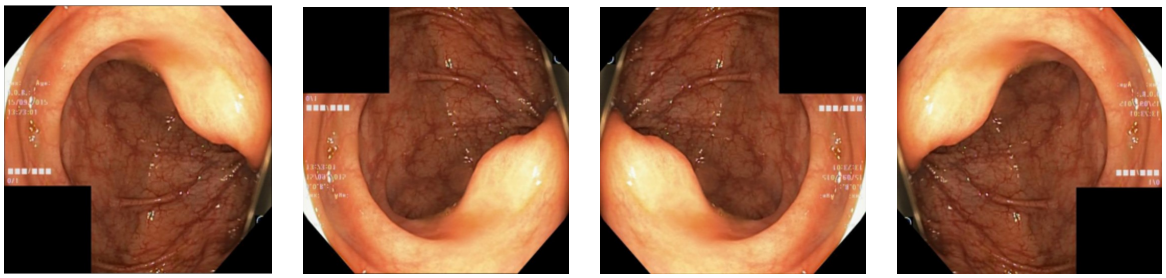
and 1. This helps to prevent exploding or vanishing gradient problems in gradient-based optimization algorithms. we divided the images by the maximum pixel value which is 255.

- **Image Size Re-scaling** Involves changing the dimensions of an image, either by reducing or increasing its size. We resized all image resolutions to a consistent 256x256 size to ensure uniformity. This resizing helps save computation and reduce training time, as downsized images have fewer pixels, requiring less memory and computation for processing.

4.1.3 Data Augmentation

Data augmentation increases training data diversity without collecting new data, which is especially useful in tasks like polyp segmentation where annotated data is scarce.Through the use of different image transformations on the available data will lead to the model being generalize to unseen samples and making more accurate segmentations.

A random combination of horizontal and vertical flips, 90-degree rotations, and image transpose, that was applied to the 880 training images, doubling the training set to 1760 images.



Original Vertical Flip 90°Rotation Horizontal Flip

Figure 4.6: Different augmentation combination for 1 sample

4.1.4 Performance Evaluation Metrics

The assessment of medical image segmentation algorithms is vital, as their performance directly influences diagnosis and treatment decisions. Evaluation metrics, a set of numerical measurements are used to evaluate the final performance of the model. [100].

4.1.4.1 Dice Coefficient

The Dice coefficient [101] is the most commonly used metric in image segmentation. It provides a measure of similarity between two sets A and B as shown in figure 4.7. In our case the predicted and ground truth segmentations. Dice coefficient is the ratio of two times the intersection divided by the area of the union. The coefficient ranges from 0 to 1, where 1 indicates a higher degree of similarity (total overlap) and 0 indicates a lower degree of similarity (no overlap).

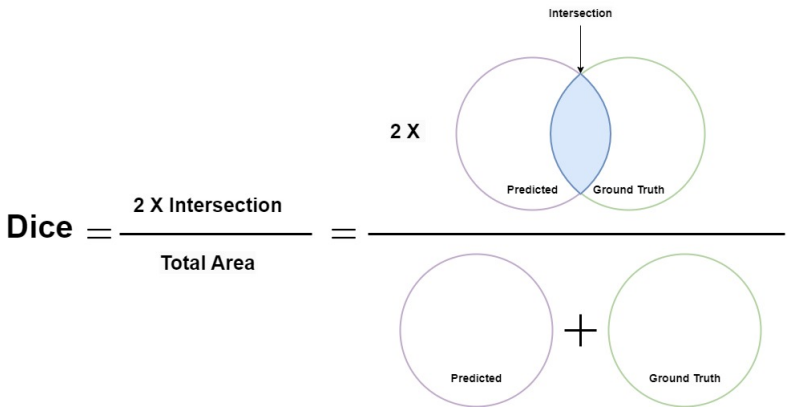


Figure 4.7: The formula to calculate Dice score

4.1.4.2 Intersection Over Union (IoU)

Similar to the Dice coefficient the Intersection over union (IoU) or Jaccard index, measures how much the ground truth (GT) mask and the predicted segments overlap [102]. The IoU is calculated as the area of intersection divided by the area of the union of the predicted and ground truth segments. The greater the overlap, the higher the IoU value. Fig 4.8 provides an illustration of how IoU is calculated.

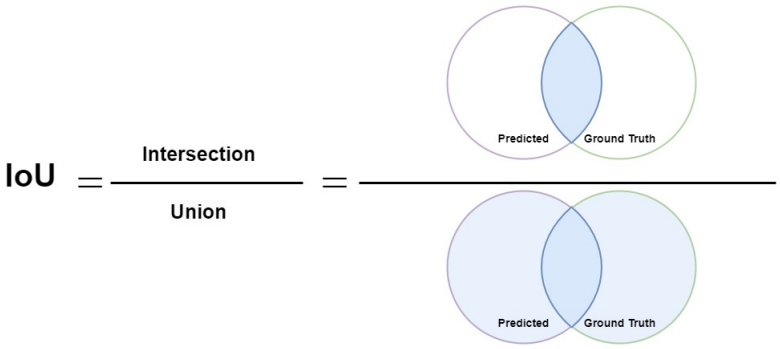


Figure 4.8: The formula to calculate IoU

4.1.4.3 Confusion Matrix

A confusion matrix is an $n \times n$ table where each row represents actual classes and each column represents predicted classes (or vice versa) [103]. Specifically, in polyp segmentation (a binary classification task), the instances for each pixel are classified as either polyp(positive) or non-polyp(negative). The confusion matrix summarizes the performance of a classification model and offers insights into potential areas for improvement. The structure of a confusion matrix for a binary classification task is displayed in the following fig 4.9

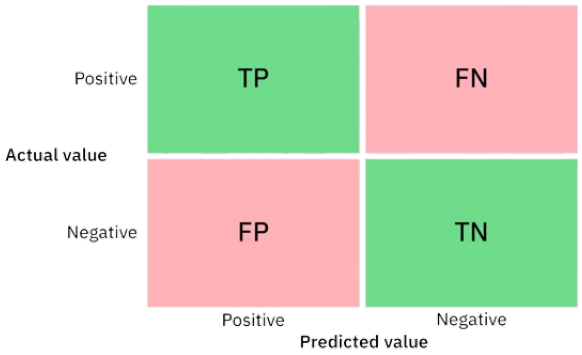


Figure 4.9: Confusion Matrix [28]

- **True Positives(TP):** Number of pixels correctly classified as polyp
- **True Negatives(TN):** Number of pixels correctly classified as non-polyp (background)
- **False Positives(FP):** Pixels are inaccurately classified as a positive class but in actuality, they are a negative class.
- **False Negatives(FN):** Pixels are inaccurately classified as a negative class but in actuality, they are a positive class.

There are sets of metrics derived from the confusion matrix, each metric provides further interpretation for the model's performance. These metrics are detailed in the following sections.

4.1.4.4 Accuracy

assesses the overall performance of the model, or in other words the ratio of properly classified samples to total samples. It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1.1}$$

4.1.4.5 Recall (Sensitify or True Positive Rate)

it measures the ability of the model to cover all actual positive samples.it is the ratio of correctly predicted positive observations to the sum of true positive (TP) and false negative (FN) instances.

$$Recall = \frac{TP}{TP + FN} \tag{4.1.2}$$

4.1.4.6 Precision

captures how accurate the model predicting positive samples by calculating the ratio of correctly predicted positive observations to the total predicted positives. It is given by:

$$Precision = \frac{TP}{TP + FP} \tag{4.1.3}$$

4.1.4.7 Specificity

it is the opposite of Recall primarily capturing how well the model identifies the actual negative samples. It is determined as:

$$Recall = \frac{TN}{TN + FP} \tag{4.1.4}$$

4.1.4.8 F1-score

The F1-Score is a fusion of two metrics Precision and Recall it provides a balance between Precision and Recall and it is useful for unbalanced classes. Calculated as:

$$F1\text{-}Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.1.5}$$

4.2 Implementation Details

The experiments were conducted on GPU P100 which has 16GB of memory and 13GB of RAM, this was provided by Kaggle in Jupyter notebooks, as well the model architecture was implemented using the latest Pytorch library. Pytorch is a framework for building deep-learning models that is compatible with the Python programming language.

4.3 Experiments and Results

In this section, we started the evaluation of our proposed approach through a series of experiments. The first experiment assesses the baseline performance of the new lightweight ResNet18 encoder without the intervention of any prompts. Subsequently, the second experiment is designed to evaluate the full potential of our proposed architecture when provided with ground truth (GT) prompts. this experiment serves as an upper bound, allowing us to understand the maximum achievable performance when the system is given perfect guidance. Before the final experiment, we conduct an evaluation to assess the proposed architecture with our added zero-shot prompting technique. This experiment focuses on the practical performance of the complete system.

Furthermore, we explore the relationship between the number of training examples and the quality of generated prompts. By varying the size of the training data used for the prompt generation, these experiments aim to determine the data requirements for creating effective prompts, ultimately shedding light on the data efficiency of our prompting approach. Through this comprehensive evaluation framework, we aim to gain a thorough understanding of our proposed method's strengths and limitations.

4.3.1 Training Process

In all conducted experiments the Dice loss was used in training with 150 epochs, and the batch size was set as 20 for the first experiment and the remaining experiments the batch size was 1 because SAM not supporting yet using prompts as batchs, moreover, the optimizer used on training is NAdam with learning rate 1.5e-4. A threshold of 0.5 was set to binaries the predicted segmentation for model evaluation.

4.3.1.1 Dice Loss

The Dice loss is a loss function used for segmentation tasks to measures the overlap between the predicted segmentation mask and the ground truth mask [104]. Defined as:

$$\boldsymbol{DiceLoss = 1 - DiceCoefficient} \tag{4.3.1}$$

4.3.2 Experiment 1: model performance without prompts

In this experiment the obtained results demonstrated promising performance, showcasing the effectiveness of the proposed approach. The model achieved 78.4% Dice, 64.7% IoU, furthermore Evaluating the model's performance in terms of precision and recall revealed a precision of 78.3% and a recall of 79%. These metrics suggest that the model strikes a balanced trade-off between correctly identifying polyp pixels while minimizing false positives and false negatives. The training results can be seen in the figure 4.10

Notably, the model achieved an overall accuracy of 93.4% with only **6.91 M** parameter and it is essential to consider that the dataset used for the experiments is highly imbalanced data but despite this challenging scenario, the model attained an F1-score of 78.49% demonstrating its capability to perform well on imbalanced data.

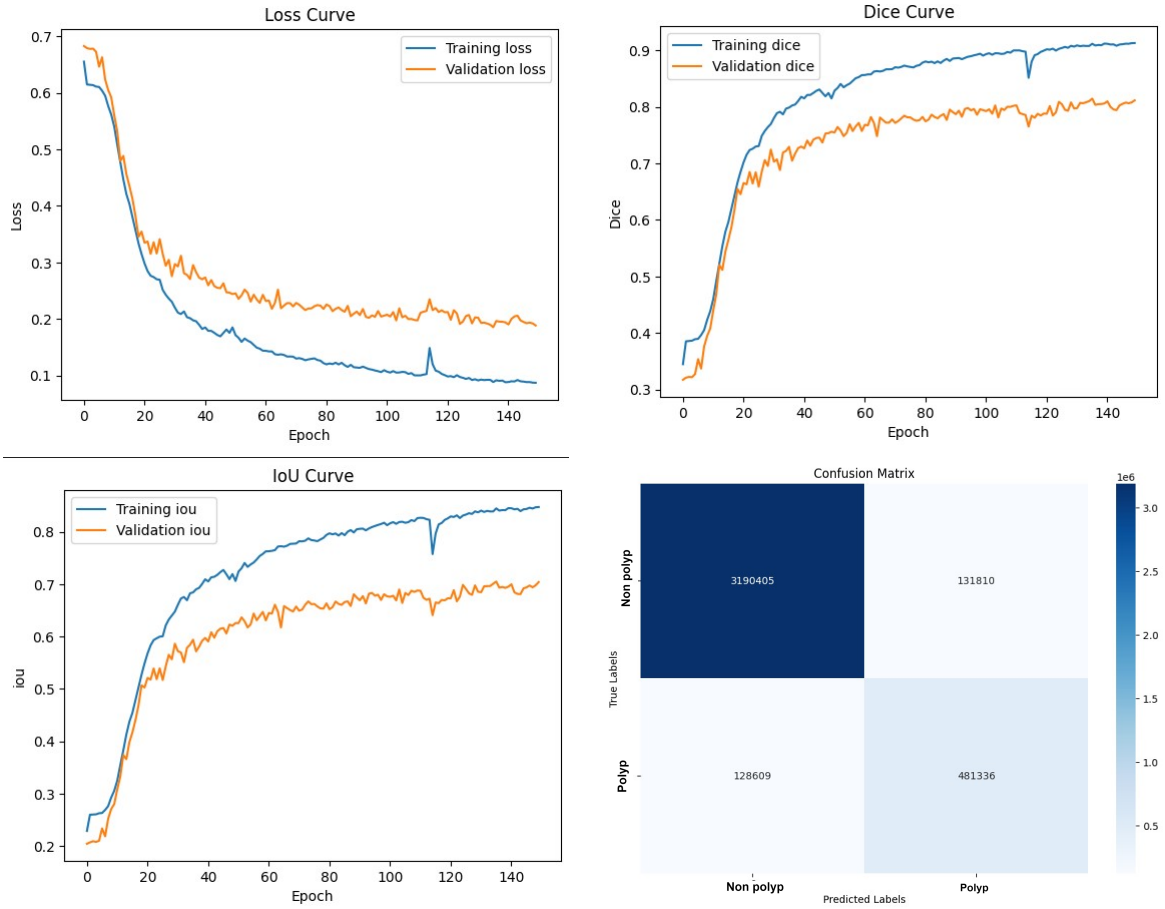


Figure 4.10: The results of the experiment 1

We can notice that learning loss and validation during learning decrease proportionally, accompanied by a proportional increase in model performance, the validation dice coefficient reached 79%. which means that our model can achieve good results without the need for prompt. We provided a comparison between different CNN, transformer, hybrid based deep learning models for polyp segmentation that were evaluated in the Kvasir-SEG dataset. We can see in figure 4.11 the predicted mask for some colonoscopy images:

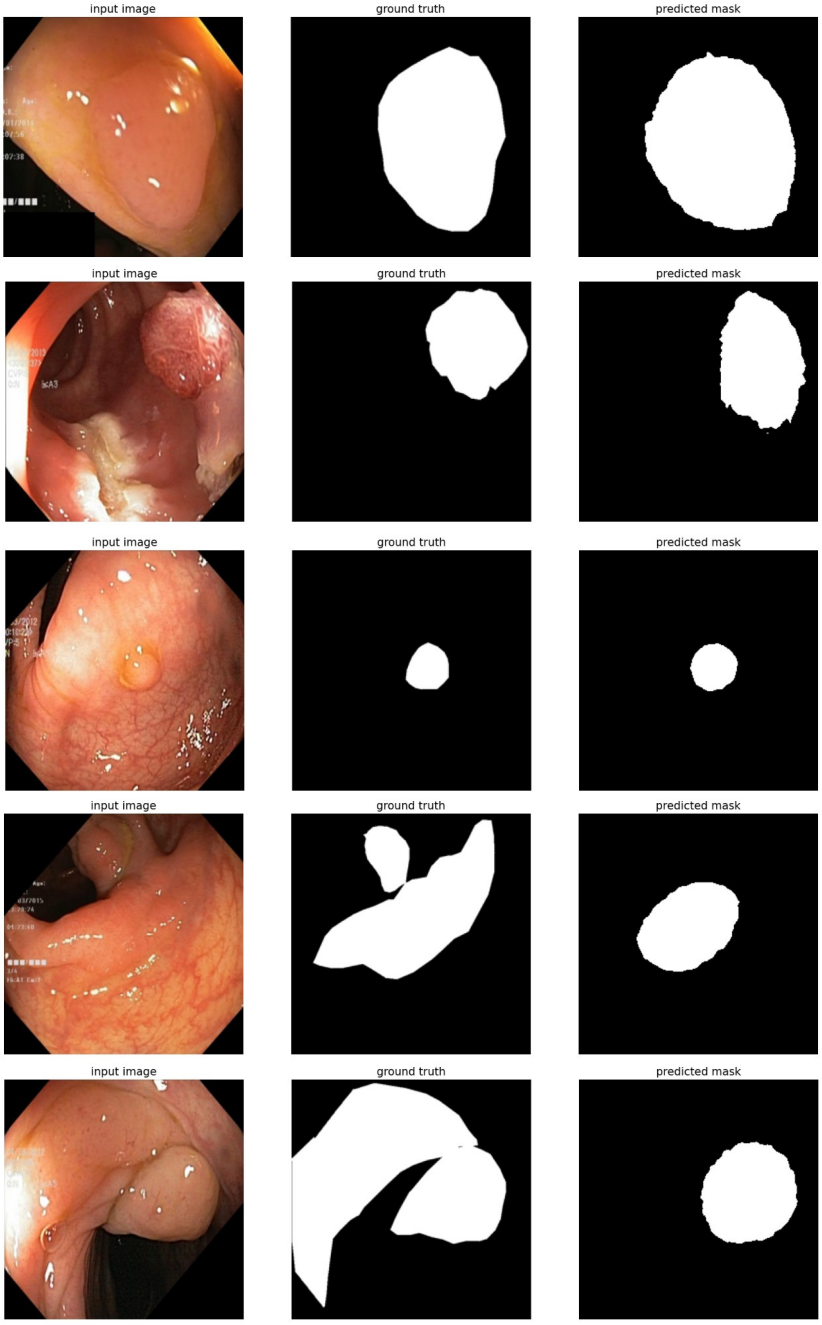


Figure 4.11: The predicted segmentation of the proposed method without prompts

4.3.3 Experiment 4: Model With Yolov8 as prompting technique

We tried to use the pre-trained Yolov8 for zero-shot object detection without any training and we found it not effective and couldn't predict boxes for polyps, then we trained it with a few examples 10,20,40,50 samples, and the results were improved for some images as it is shown in figure 4.12, but the majority were not good. so we fine-tuned it using 200 images from the training set for 500 epochs using an SGD optimizer with a learning rate of 1e-3 and got good detection results. then we use it as a prompt generator for our model we select the best box by score and we set the box for images that Yolo does not predict any box for it with a box around all the images, the results of the training process for 125 epochs can be seen in the figure 4.13

We can notice that the model converged directly due to the accuracy of the prompts that were extracted using Yolov8, while its results were not very accurate (achieve 67% Dice and 56% IoU) due to some images in which Yolov8 was not able to extract the boxes accurately which leads to distortion of the model and limits its generalizability, some examples of results shown in the figure 4.14.

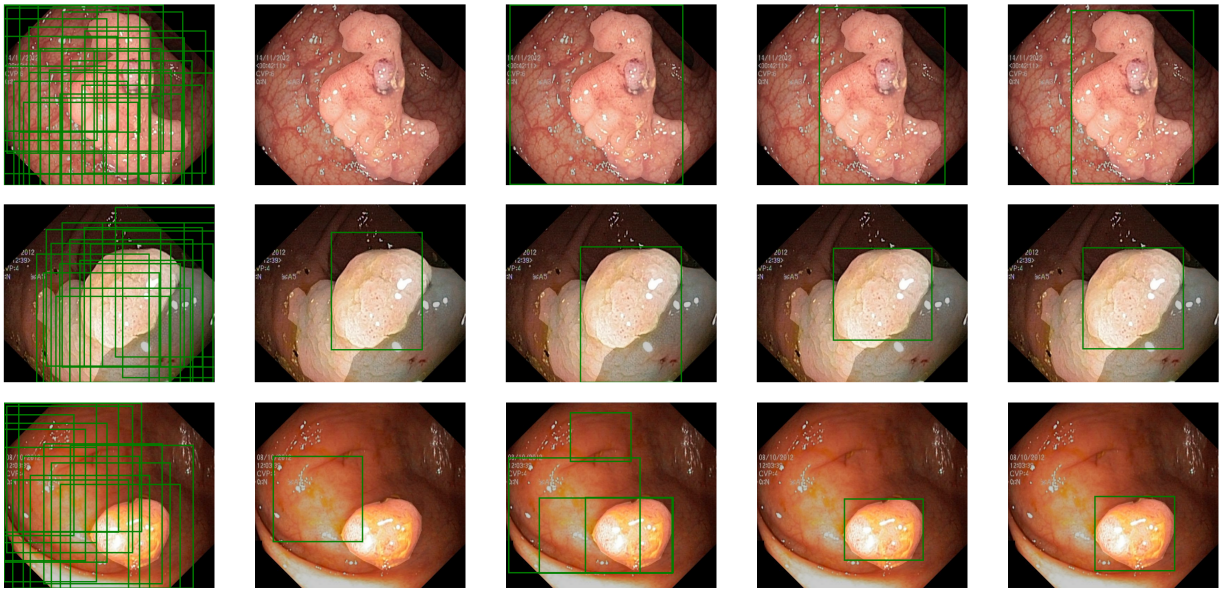
The table **??** summarizes the results of our proposed method compared to the other SOTA in terms of the number of parameters and the achieved Dice and IoU scores, the obtained results from .

4.4 Discussion

The initial experiment evaluated the lightweight nature of our proposed Segment Anything Model (SAM) approach for polyp segmentation. Despite its lightweight architecture comprising only 6.91 million parameters in total, the model demonstrated promising performance without prompt that is comparable to other state-of-the-art transformer models, such as TransUNet, AutoSAM, and even the fine-tuned full SAM (PolypSAM), also these promising results highlight the potential of the proposed lightweight model, for deployment in resource-constrained environments and data-limited scenarios. When the lightweight SAM is prompted with effective prompts(GT-prompts) as shown in the second experiment, and achieves 81% dice same as Unet with only 6.91 million parameters. However, while the model exhibited competitive performance overall, it faced challenges when trying to segment different polyps within the same image. In such scenarios, the model's accuracy decreased, indicating that further improvements are necessary to enhance its capability to manage multi-instance cases effectively.

To address this limitation, the proposed zero-shot object detection combined with K-means bounding box refinement may offer a solution. By generating multiple bounding boxes, the approach aims to segment multiple polyps accurately within a single image.

The suggested zero-shot object detection still requires more improvements. GroundingDino's text input prompt engineering could enhance the generated bounding boxes. Combining other types of prompts, such as point or mask with a bounding box, could improve the obtained results and possibly improve segmentation performance as a whole.



10 examples 20 examples 40 examples 50 examples 200 examples

Figure 4.12: The results of yolov8 trained on 10,20,40,50 and 200 examples

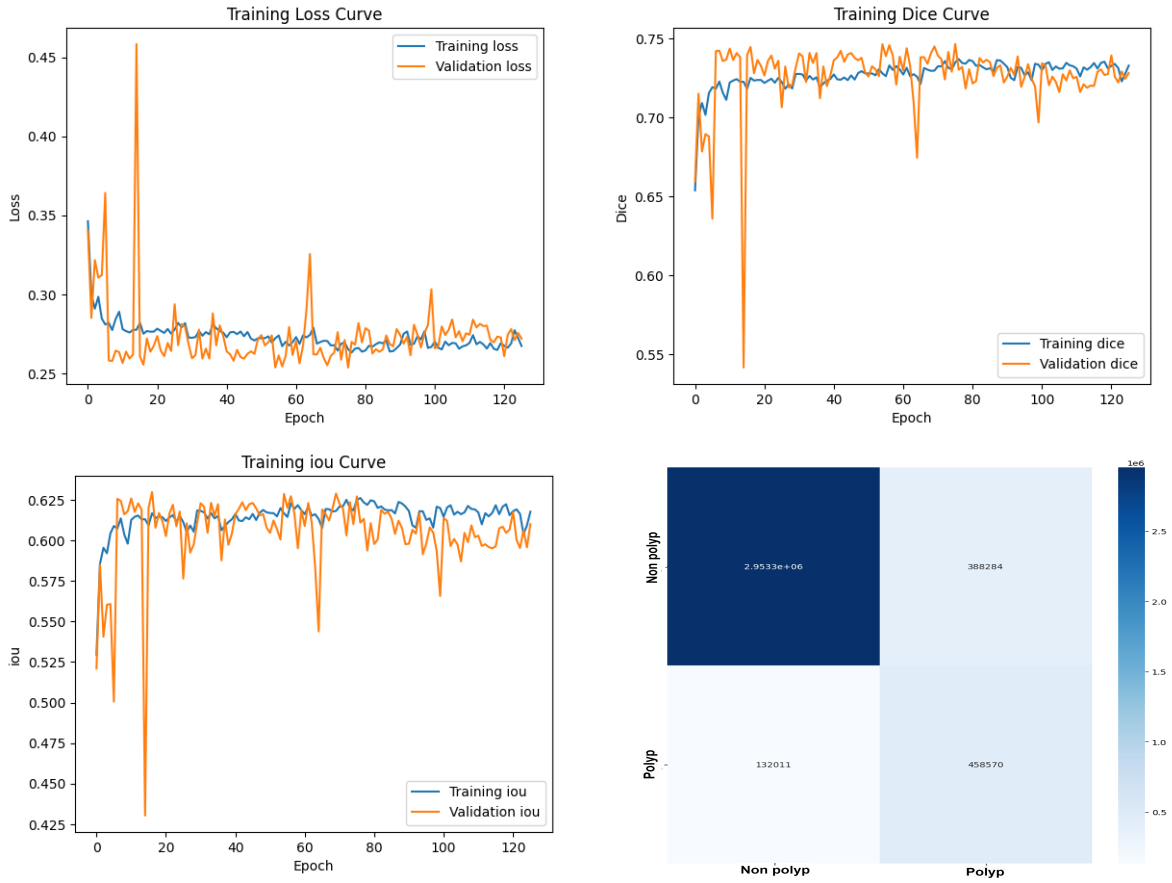


Figure 4.13: The results of the experiment 4

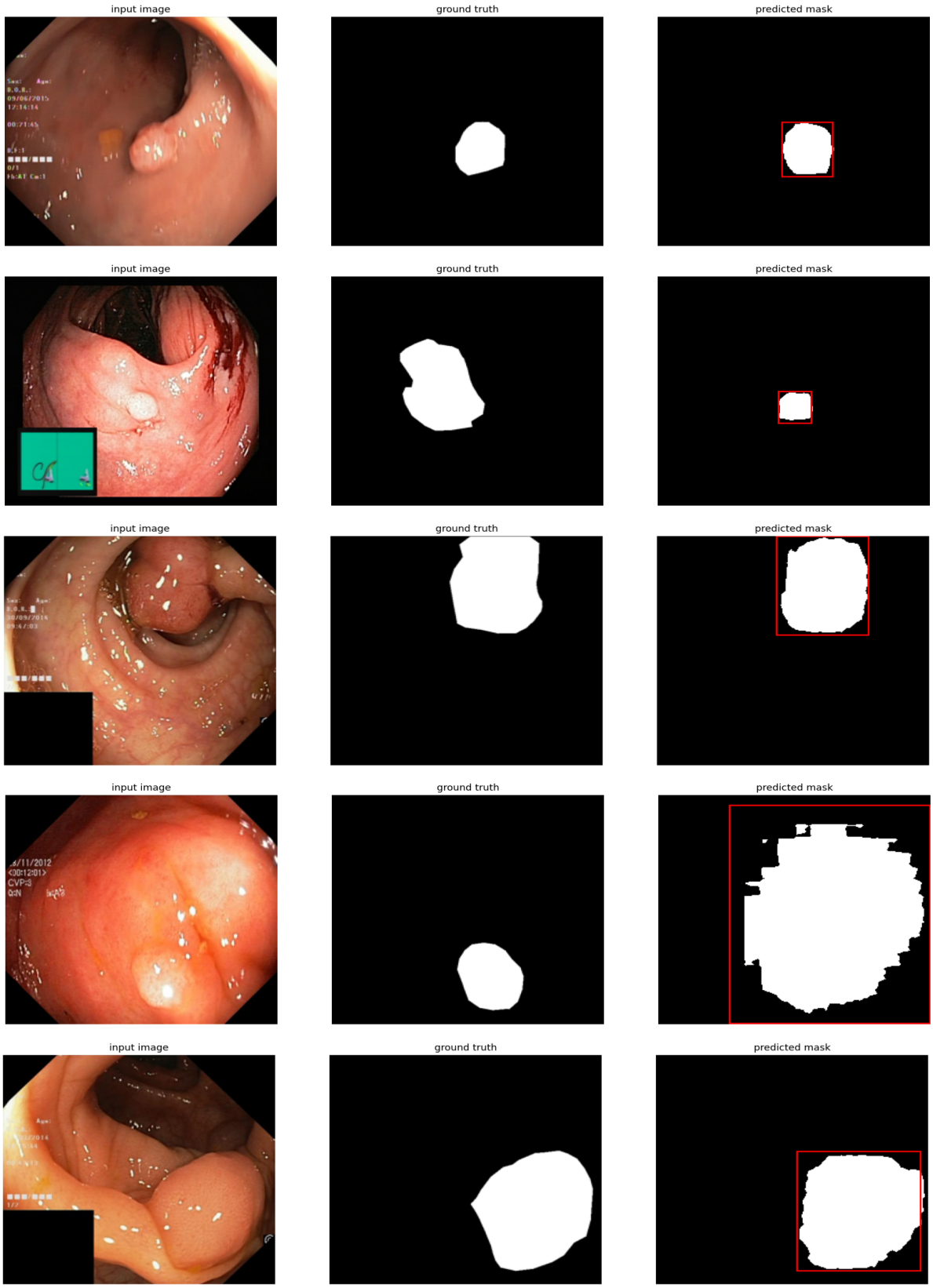


Figure 4.14: The predicted segmentation of the proposed method with yolov8 boxes

General Conclusion

In this thesis, we concentrated our efforts on developing a lightweight version of the Segment Anything Model (SAM) with a zero-shot prompting strategy, ultimately adapting SAM for polyp segmentation in data-limited settings. We introduced a lightweight SAM by replacing its vision transformer encoder with a lightweight pre-trained ResNet18 model. This step significantly reduced the number of parameters from 91M to 3M, preventing the risk of overfitting on limited data and enabling the deployment of a lightweight model on various resource-constrained devices. Additionally, this strategy minimizes the computational resources required for training the model.

The zero-shot prompting strategy generates bounding boxes for polyps using a text description by employing a pre-trained object detection model called GroundingDINO, followed by K-means clustering for bounding box refinement. The resulting bounding box serves as a prompt for the Segment Anything Model (SAM), allowing it to generalize to unseen data without retraining. Incorporating this bounding box as a guide focuses SAM's attention on the desired object, leading to improved segmentation accuracy and faster convergence.

Next, we provided the used dataset to evaluate the proposed method, along with dataset analysis and all the preprocessing steps undertaken. with outlining the evaluation metrics employed to asses the performance of the proposed model. Finally, we presented the obtained results of different experiment settings.

The obtained results were promising in the context of a lightweight SAM without prompts, and the results were competitive when prompted with ground truth prompts. The zero-shot prompting strategy also yielded promising results, this prompting strategy could be an efficient fine-tuning strategy in limited data settings. Additionally, the prompting strategy can be applied to other medical image segmentation tasks, highlighting its adaptability.

In conclusion, overall the application of SAM for medical image segmentation is still yet relatively new research area with significant potential for further exploration and development, ultimately enhancing the quality of human life by improving diagnostic accuracy and facilitating better treatment planning.

Bibliography

[1] Li Zhang, Youwei Liang, Ruiyi Zhang, Amirhosein Javadi, and Pengtao Xie. Blo-sam: Bi-level optimization based overfitting-preventing finetuning of sam, 2024.

[2] Khalid Abdus Sattar. Tadoc : Tool for automated detection of oral cancer. *International Journal of Advanced Computer Science and Applications*, 11:506–513, 04 2020.

[3] Hireterra. Machine Learning in Computer Vision, 2022. [Accessed on 2024].

[4] Tiwari S. Khurana M. Arya K.V Dang, N. *Recent Advancements in Medical Imaging: A Machine Learning Approach*. Springer, Singapore, 2021.

[5] Cong T Nguyen, Nguyen Van Huynh, Nam H Chu, Yuris Mulya Saputra, Dinh Thai Hoang, Diep N Nguyen, Quoc-Viet Pham, Dusit Niyato, Eryk Dutkiewicz, and Won-Joo Hwang. Transfer learning for wireless networks: A comprehensive survey. *Proceedings of the IEEE*, 110(8):1073–1115, 2022.

[6] Qiming Zhang, Haoyi Yu, Martina Barbiero, Baokai Wang, and Min Gu. Artificial neural networks enabled by nanophotonics. *Light: Science & Applications*, 8(1):42, 2019.

[7] Nidhi Sahai/ANALYTIXLABS. Convolutional neural networks – definition, architecture, types, applications, and more. `https://www.analytixlabs.co.in/blog/convolutional-neural-network/`.

[8] Maria Vakalopoulou, Stergios Christodoulidis, Ninon Burgos, Olivier Colliot, and Vincent Lepetit. Deep learning: basics and convolutional neural networks (cnns). *Machine Learning for Brain Disorders*, pages 77–115, 2023.

[9] Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Poornima Singh Thakur, Pritee Khanna, Tanuja Sheorey, and Aparajita Ojha. Explainable vision transformer enabled convolutional neural network for plant disease identification: Plantxvit. *arXiv preprint arXiv:2207.07919*, 2022.

[12] Lichun Yu and Jinqing Liu. Face recognition based on deep learning of small data set. *Journal of Physics: Conference Series*, 1624:052004, 10 2020.

[13] Saleh Albahli and Tahira Nazir. Ai-centernet cxr: An artificial intelligence (ai) enabled system for localization and classification of chest x-ray disease. *Frontiers in Medicine*, 9, 08 2022.

[14] Kalyani Kadam, Swati Ahirrao, and Ketan Kotecha. Efficient approach towards detection and identification of copy move and image splicing forgeries using mask r-cnn with mobilenet v1. *Computational Intelligence and Neuroscience*, 2022:1–21, 01 2022.

[15] Thi-Hai-Binh Nguyen, Eunsoo Park, Xuenan Cui, Van Nguyen, and Hakil Kim. fpadnet: Small and efficient convolutional neural network for presentation attack detection. *Sensors*, 18:2532, 08 2018.

[16] Jialin Peng and Ye Wang. Medical image segmentation with limited supervision: A review of deep network models. *IEEE Access*, 9:36827–36851, 2021.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[18] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[19] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[20] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *ECCV Workshops*, 2021.

[21] Olivier Petit, Nicolas Thome, Clement Rambour, Loic Themyr, Toby Collins, and Luc Soler. U-net transformer: Self and cross attention for medical image segmentation. In *Machine Learning in Medical Imaging: 12th International Workshop, MLMI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 12*, pages 267–276. Springer, 2021.

[22] Chunhui Zhang, Li Liu, Yawen Cui, Guanjie Huang, Weilin Lin, Yiqian Yang, and Yuehong Hu. A comprehensive survey on segment anything model for vision and beyond. *arXiv preprint arXiv:2305.08196*, 2023.

[23] Kaidong Zhang and Dong Liu. Customized segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.13785*, 2023.

[24] Li Zhang, Youwei Liang, and Pengtao Xie. Blo-sam: Bi-level optimization based overfitting-preventing finetuning of sam. *arXiv preprint arXiv:2402.16338*, 2024.

[25] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15:1–9, 2024.

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[27] Zhenyu Wu, Fengmao Lv, Chenglizhao Chen, Aimin Hao, and Shuo Li. Colorectal polyp segmentation in the deep learning era: A comprehensive survey. *arXiv preprint arXiv:2401.11734*, 2024.

[28] Create a confusion matrix with python, use scikit-learn to create a confusion matrix for a simple binary classification problem. `https://developer.ibm.com/tutorials/awb-confusion-matrix-python/`.

[29] Long Wen, Liang Gao, and Xinyu Li. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49:136–144, 2019.

[30] Leng Hui. Kang, Tai-Jiang Mu, and Guoping Zhao. Hierarchical transformer-based siamese network for related trading detection in financial market. *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2023.

[31] Oscar et al Hernandez Dominguez. Stage iv colorectal cancer management and treatment. *Journal of clinical medicine*, 12,5 2072, 6 Mar. 2023.

[32] Maciej Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study, 04 2023.

[33] Zihan Zhong, Zhiqiang Tang, Tong He, Haoyang Fang, and Chun Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model, 2024.

[34] Jiaxing Huang, Kai Jiang, Jingyi Zhang, Han Qiu, Lewei Lu, Shijian Lu, and Eric Xing. Learning to prompt segment anything models, 2024.

[35] Mohammed Amine Merzougui and Ahmad El Allaoui. Region growing segmentation optimized by evolutionary approach and maximum entropy. In *ANT/EDI40*, 2019.

[36] Akmal Shafiq Badarul Azam, Aminah Abdul Malek, Aida Sharmaine Ramlee, Nur Diana Syahira Muhammad Suhaimi, and Norlyda Mohamed. Segmentation of breast microcalcification using hybrid method of canny algorithm with otsu thresholding and 2d wavelet transform. *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 91–96, 2020.

[37] Pei Yang, Wei Song, Xiaobing Zhao, Rui Zheng, and Letu Qingge. An improved otsu threshold segmentation algorithm. *Int. J. Comput. Sci. Eng.*, 22:146–153, 2020.

[38] Yu Wang, DR Chen, ML Shen, and Ge Wu. Watershed segmentation based on morphological gradient reconstruction and marker extraction. *Journal of image and graphics*, 13(11):2176–2180, 2008.

[39] Sa Yoganathan and Rui Zhang. Segmentation of organs and tumor within brain magnetic resonance images using k-nearest neighbor classification. *Journal of Medical Physics*, 47:40–9, 04 2022.

[40] Alena Shamsheyeva and Arcot Sowmya. Tuning kernel function parameters of support vector machines for segmentation of lung disease patterns in high-resolution computed tomography images. In *SPIE Medical Imaging*, 2004.

[41] S. Akshay and Puttagunta Sree Apoorva. Segmentation and classification of fmm compressed retinal images using watershed and canny segmentation and support vector machine. *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 1035–1039, 2017.

[42] Srikumaran R. Enhancing brain tumor diagnosis: A synergistic approach with support vector machines and decision trees for improved detection and segmentation. *International Journal for Research in Applied Science and Engineering Technology*, 2024.

[43] Noor Salah Hassan, Adnan Mohsin Abdulazeez, Diyar Qader Zeebaree, and Dathar Abas Hasan. Medical images breast cancer segmentation based on k-means clustering algorithm: A review. *Asian Journal of Research in Computer Science*, 2021.

[44] Karim M. Aljebory, Thabit Sultan Mohammed, and Mohammed U. Zainal. Enhanced image segmentation: Merging fuzzy k-means and fuzzy c-means clustering algorithms for medical applications. *Computer Science and Information Technology*, 2021.

[45] Martin Müller, Marie Stiefel, Björn Bachmann, Dominik Britz, and Frank Mücklich. Overview: Machine learning for segmentation and classification of complex steel microstructures. *Metals*, 2024.

[46] Evgin Göçeri, Esther Durá, and Melih Günay. Review on machine learning based lesion segmentation methods from brain mr images. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 582–587, 2016.

[47] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin P. Murphy, and Alan Loddon Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:834–848, 2016.

[48] Mohand Tuffaha and M. Rosario Perello Marin. Artificial intelligence definition, applications and adoption in human resource management: a systematic literature review. *International Journal of Business Innovation and Research*, 2023.

[49] Sheng He, Rina Bao, Jingpeng Li, Patricia Ellen Grant, and Yangming Ou. Accuracy of segment-anything model (sam) in medical image segmentation tasks. *ArXiv*, abs/2304.09324, 2023.

[50] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, Sijing Liu, Haozhe Chi, Xindi Hu, Kejuan Yue, Lei Li, Vicente Grau, Deng-Ping Fan, Fajin Dong, and Dong Ni. Segment anything model for medical images? *ArXiv*, 2024.

[51] Roohollah Aslanzadeh, Kazem Qazanfari, and Mohammad Rahmati. An efficient evolutionary based method for image segmentation. *arXiv preprint arXiv:1709.04393*, 2017.

[52] What is medical image analysis? `https://www.mathworks.com/discovery/medical-image-analysis.html/`.

[53] Aurélien Géron. Hands-on machine learning with scikit-learn, keras, and tensorflow: concepts. *Aurélien Géron-Google Kitaplar, yy https://books. google. com. tr/books*, 2019.

[54] Andrea Lodi and Giulia Zarpellon. On learning and branching: A survey. Report, April 2017.

[55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[56] Frank Emmert-Streib and Matthias Dehmer. Taxonomy of machine learning paradigms: A data-centric perspective. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(5):e1470, 2022.

[57] Wang Yaqing, Yao Quanming, T Kwok James, and M Ni Lionel. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys*, 53(3):1–34, 2020.

[58] Mcculloch-pitts neuron — mankind's first mathematical model of a biological neuron. `https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1/`.

[59] Tianfeng Chai and Roland R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7:1247–1250, 2014.

[60] Juan Terven, Diana-Margarita Cordova-Esparza, Alfonzo Ramirez-Pedraza, and Edgar Chávez Urbiola. Loss functions and metrics in deep learning. a review, 07 2023.

[61] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[62] Asifullah Khan, Zunaira Rauf, Anabia Sohail, Abdul Rehman, Hifsa Asif, Aqsa Asif, and Umair Farooq. A survey of the vision transformers and its cnn-transformer based variants. *arXiv preprint arXiv:2305.09880*, 2023.

[63] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[64] Yifan Xu, Huapeng Wei, Minxuan Lin, Yingying Deng, Kekai Sheng, Mengdan Zhang, Fan Tang, Weiming Dong, Feiyue Huang, and Changsheng Xu. Transformers in computational visual media: A survey. *Computational Visual Media*, 8:33–62, 2022.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[66] Yi-Lun Liao, Sertac Karaman, and Vivienne Sze. Searching for efficient multi-stage vision transformers. *arXiv preprint arXiv:2109.00642*, 2021.

[67] Edoardo Debenedetti and Carmela Troncoso—EPFL. *Adversarially robust vision transformers*. PhD thesis, Master's thesis, Swiss Federal Institue of Technology, Lausanne (EPFL), 2022.

[68] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

[69] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[70] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, abs/1704.04861, 2017.

[71] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *ArXiv*, abs/1602.07360, 2016.

[72] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

[73] Chetan Tulasigeri and M. Irulappan. An advanced thresholding algorithm for diagnosis of glaucoma in fundus images. *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1676–1680, 2016.

[74] V. Sivakumar and V. Murugesh. A brief study of image segmentation using thresholding technique on a noisy image. *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pages 1–6, 2014.

[75] Bickey Kumar, Vansh Kedia, Rohan P. Raut, Sakil Ansari, and Anshul Shroff. Evaluation and comparative study of edge detection techniques. 2020.

[76] Shouvik Chakraborty. An advanced approach to detect edges of digital images for image segmentation. 2020.

[77] Akshay P. Vartak and Dr V R Mankar. Morphological image segmentation analysis. 2013.

[78] Wenjian Yao, Jiajun Bai, Wei Liao, Yuheng Chen, Mengjuan Liu, and Yao Xie. From cnn to transformer: A review of medical image segmentation models. *Journal of Imaging Informatics in Medicine*, pages 1–19, 2024.

[79] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. pages 3431–3440, 06 2015.

[80] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S...*, 11045:3–11, 2018.

[81] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, M. J. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas. *ArXiv*, abs/1804.03999, 2018.

[82] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15:749–753, 2017.

[83] Nastaran Enshaei, Moezedin Javad Rafiee, and Farnoosh Naderkhani. A generalization enhancement approach for deep learning segmentation models: Application in covid-19 lesion segmentation from chest ct slices. *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1362–1366, 2022.

[84] Neng Zhou, Hairu Wen, Yi Wang, Yang Liu, and Longfei Zhou. Review of deep learning models for spine segmentation. *Proceedings of the 2022 International Conference on Multimedia Retrieval*, 2022.

[85] Ling Zhang, Xiaosong Wang, Dong Yang, Thomas Sanford, Stephanie A. Harmon, Baris I Turkbey, Bradford J. Wood, Holger R. Roth, Andriy Myronenko, Daguang Xu, and Ziyue Xu. Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE Transactions on Medical Imaging*, 39:2531–2540, 2020.

[86] Zihan Li, Yunxiang Li, Qingde Li, Puyang Wang, Dazhou Guo, Le Lu, Dakai Jin, You Zhang, and Qingqi Hong. Lvit: Language meets vision transformer in medical image segmentation. *IEEE Transactions on Medical Imaging*, PP:12, 06 2023.

[87] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[88] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[89] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[90] Dongjie Cheng, Ziyuan Qin, Zekun Jiang, Shaoting Zhang, Qicheng Lao, and Kang Li. Sam on medical images: A comprehensive study on three prompt modes. *arXiv preprint arXiv:2305.00035*, 2023.

[91] Qi Wu, Yuyao Zhang, and Marawan Elbatel. Self-prompting large vision models for few-shot medical image segmentation. In *MICCAI Workshop on Domain Adaptation and Representation Transfer*, pages 156–167. Springer, 2023.

[92] Risab Biswas. Polyp-sam++: Can a text guided sam perform better for polyp segmentation? *arXiv preprint arXiv:2308.06623*, 2023.

[93] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[94] Shurong Chai, Rahul Kumar Jain, Shiyu Teng, Jiaqing Liu, Yinhao Li, Tomoko Tateyama, and Yen-wei Chen. Ladder fine-tuning approach for sam integrating complementary network. *arXiv preprint arXiv:2306.12737*, 2023.

[95] Yuheng Li, Mingzhe Hu, and Xiaofeng Yang. Polyp-sam: Transfer sam for polyp segmentation. In *Medical Imaging 2024: Computer-Aided Diagnosis*, volume 12927, pages 759–765. SPIE, 2024.

[96] Zhaozhi Xie, Bochen Guan, Weihao Jiang, Muyang Yi, Yue Ding, Hongtao Lu, and Lei Zhang. Pa-sam: Prompt adapter sam for high-quality image segmentation. *arXiv preprint arXiv:2401.13051*, 2024.

[97] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[98] ENCORD BLOG. Meta ai's segment anything model (sam) explained: The ultimate guide. `https://encord.com/blog/segment-anything-model-explained/`.

[99] Kvasir-seg dataset. `https://datasets.simula.no/kvasir-seg/`.

[100] Dominik Müller, Iñaki Soto-Rey, and Frank Kramer. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*, 15(1):210, 2022.

[101] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[102] intersection-over-union-iou. `https://viso.ai/computer-vision/intersection-over-union-iou/`.

[103] segmentationconfusionmatrix. `https://www.mathworks.com/help/vision/ref/segmentationconfusionmatrix.html`.

[104] Carole Helene Sudre, Wenqi Li, Tom Kamiel Magda Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City, QC,...*, 2017:240–248, 2017.