Deep Learning for Automated Polyp Detection and Localization in Colonoscopy

Amita Patra

Thesis submitted for the degree of
Master in Applied Computer and Information Technology (ACIT)
30 credits

Department of Computer Science
Faculty of Technology, Art and Design

OSLO METROPOLITAN UNIVERSITY

Spring 2022

Deep Learning for Automated Polyp Detection and Localization in Colonoscopy

Amita Patra

Deep Learning for Automated Polyp Detection and Localization in Colonoscopy

http://www.oslomet.no/

Printed: Oslo Metropolitan University

Abstract

Gastrointestinal (GI) tract comprises organs from mouth to anus. Multiple diseases can occur in the GI tract. Among the different diseases in the digestive system, the most commonly found cancer in the gastrointestinal tract are oesophagus cancer, stomach cancer and colorectal cancer (CRC). Amongst them, CRC is the third most commonly caused cancer in terms of incidence and the second most commonly leading cause of cancer-related death worldwide. Norway specifically has the highest occurrence of colon cancer worldwide. Prior detection is a crucial factor to improve chances of survival, prognosis and timely treatment. Recently, there has been significant progress made in the field of medical image analysis through the development of Computer-aided diagnosis systems. CADx systems utilizing machine learning algorithms, specifically deep learning methods have produced excellent object detection models for automatic detection of abnormalities in medicine. In this respect, this thesis explores the possibility of implementing recent state-of-the-art deep learning models for automated polyp detection. The You Only Look Once (YOLO) family of models have demonstrated good results for object detection tasks, especially YOLOv4. Lately, YOLOv5, YOLO Representation (YOLOR) and Paddle-Paddle YOLO (PP-YOLO) were released which have shown even an improved performance over the earlier version YOLOv4. These real-time detection models were used together with the Kvasir-SEG dataset and BKAI-IGH NeoSmall-Polyp that contains several GI tract images with annotations to detect CRC precursor lesions called polyps. The models were compared to previous state-of-the-art models using metrics mAP, precision and recall. The models showed improved performance for mAP value. Additionally, we also achieved real-time processing speed with our models. However, the models achieved satisfactory accuracy for detection which indicates that there is still future scope for improvement. The experimental results presented in the

study show that the presented method can be a strong benchmark for the development of an automated polyp detection system in real-world clinical applications.

Acknowledgments

First of all, I would like to thank Almighty God for showering his blessings and for giving me this opportunity to complete Masters Degree Program and successfully completing the research work.

I would like to express my sincere gratitude to my research supervisors Dr. Pål Halvorsen, Dr. Michael Riegler and Dr. Debesh Jha for their support, motivation and guidance throughout the master thesis journey. Without their valuable feedback and proper guidance in right direction, this thesis and research work would not have been completed.

I am extremely grateful to my husband for being a source of constant motivation and supporting me during tough times. A big thanks to my daughter who was cooperative throughout my master thesis. Last but not the least, I am specially thankful to my mother and my in-laws for their love and blessings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The digestive system of humans could be affected with several types diseases. In particular, the gastrointestinal (GI) tract, going from the mouth to the anus may be infected by different types of cancer such as colorectal cancer (CRC), stomach cancer and oesophagus cancer [140]. Among which CRC is one of the leading causes of cancer-related deaths among both men and women globally [102]. The Nordic countries have had an increasing trend in the CRC being diagnosed since past 60 years [16] [78]. Additionally, Norway has shown a tremendous rise in the CRC incidences in the last 50 years, and the CRC incidences in Norway are recorded the highest globally [15] [85].

The beginning of CRC starts with the non cancerous growth of the tissues along the wall of colon (large bowel) or rectum known as polyps. Most polyps are not harmful but if not treated in time could result in cancer. Therefore, early and timely detection of polyps in the GI tract is crucial in preventing spread of the disease and further complications and giving increased options for treatments. Several research studies also indicate that screening among the population improves the prognosis and can also help in reducing the incidences of CRC [62].

Colonoscopy is a golden standard for colon examination in early detection and removal of polyps. Colonoscopy is a clinical procedure that requires the colonoscope to be inserted into the body through the anus for diagnosis

which is unpleasant for the patients. However, colonoscopy requires significant amount of time and effort by the specialized physicians. Moreover, the results are highly dependent on the physician and nearly 25-28% of the polyps could be missed in a single colonoscopy [60] [17]. Alternative to traditional colonoscopy, an examination can be performed through a camera pill known as wireless capsule endoscopy. The pill needs to be swallowed by the patient and it captures images and videos of the entire gastrointestinal tract system. The recorded data is then analysed by the specialized physician in diagnosing the ailments. However, the accuracy of the results completely relies on the basis of skills, expertise and motivation of the medical personnel. Therefore, it is essential to develop an assisted diagnosis system for detection and localization of diseases from the various colonoscopic examinations which would be able to provide assistance to the medical professionals.

The recorded and stored dataset in the examinations by the hospitals could be put to use and has great potential using various artificial intelligence techniques in assisting the diagnosis of GI tract diseases through detection and localization of polyps. This would in turn provide assistance to the medical practitioners, support in reducing manual work and building comprehensive solution for improving the detection rates of polyps which otherwise could easily be missed by less qualified medical personnel.

1.2 Problem Statement

With regards to the motivation provided in the previous section, this thesis aims in exploring the potential outcomes of implementing the recently launched object detection models which have proven to show a good detection accuracy. These models would be utilized and configured for building an assisted diagnosis system based on deep learning which would be able to detect and localize the polyps in images from colonoscopic examinations. For achieving this we will work with the available dataset containing images with annotations from the real colonoscopy examinations namely Kvasir-SEG [74] and BKAI-IGH NeoPolyp-Small [29].

The research questions to be addressed in this thesis are as following:

1. Can artificial intelligence (AI) techniques such as deep learning in object detection and localization develop a concrete medical diagnosis solution which achieves maximum accuracy and correctness of the outcome in terms of detecting an localizing polyps that has comparable results when manually diagnosed?

A machine learning based computer vision algorithm requires many samples for training the models. The dataset Kvasir-SEG [74] and BKAI-IGH NeoPolyp-Small [29] contain images with annotations for bounding boxes containing polyp object will be utilized for training and testing the models.

To answer the above research questions, we define the following objectives:

- **Objective 1** Implement the training model for object detection and localization in medical images from the gastrointestinal tract.
- **Objective 2** Validate and evaluate the performance and results of the trained model on GI tract images on unseen medical images and compare it with the ground truth values.
- **Objective 3** Compare the object detection evaluation results of the trained models with the earlier state-of-the-art benchmark models.
- **Objective 4** The detection model should be expandable for solving other tasks such as polyp segmentation and classification.

1.3 Limitations and Scope

Although there are various tasks in computer vision field such as image classification, segmentation, object detection and localization, the thesis work is confined to object detection and localization tasks. The scope of this thesis is to gain insights and explore the potential of the CNN-based deep learning architecture and implement recently launched versions in the YOLO family of models namely YOLOv5 [75], YOLOR [130] and PP-YOLO [84] for polyp detection and localization. The models would be implemented with fine tuning to evaluate and compare the performance results with existing state-of-the-art models. The dataset used for running different experiments is Kvasir-SEG [74] and BKAI-IGH NeoPolyp-Small [29]. Also, the scope of this

thesis work is only to detect whether the polyp is present in the given image or not and does not specifically identify or classify the type of the polyp which is being detected. The model training and experiments are performed on the hardware of NVIDIA Tesla K80 GPU (Graphics processing Unit) which is appropriate for training model with smaller datasets as the hardware is not powerful for running experiments for large scale data.

1.4 Research Methods

AI and computer vision is a field of scientific research. Research method used in this work is the design paradigm which has been presented by the Association for Computing Machinery (ACM) Task Force in the report named Computing as a Discipline [45]. The four steps involved for addressing a given problem in the design paradigm are as follows: define the requirements; define the specifications; designing and implementation of the system; testing of the implemented system [45]. The objective is to develop and implement the model from the dataset available and to deduce statistical findings from the data. Therefore, the design paradigm would be essentially put to use in constructing object detection models. The methodology approach used in performing the various experiments for object detection and localization includes: Data preparation and pre-processing, Model initialization, Training the model, Testing the model and Model evaluation. The experiments would be performed iteratively in order to obtain improved results for detection accuracy.

1.5 Main Contributions

In this thesis work, we have shown that the newer versions of detection models in the YOLO family of models based on deep learning could be utilized effectively and easily for performing polyp detection and localization task. Also,it proves that the AI techniques involving deep learning in the field of computer vision tasks could be used extensively and has a potential in the medical research areas. The initial results obtained with basic dataset and model configuration setup depicts that there are huge prospects for further exploring the research in the medical imaging domain particularly for

disease detection in gastrointestinal tract. The method YOLOv5 and PP-YOLO achieved fairly good detection accuracy with precision of 0.781 and 0.885 respectively for unseen test dataset. However, the detection accuracy for YOLOR, PP-YOLO model was not satisfactory having precision value of 0.442 despite having good training and evaluation results. The speed of detection for all the three models was adequate with YOLOv5 [75] having FPS=106.38, YOLOR [130] having FPS=76.62 and PP-YOLO [84] having FPS=45.04. The YOLOv5 model achieved striking FPS value of 121.96. The mean average precision values for the models also showed good results having 0.721, 0.628 and 0.888 for YOLOv5, YOLOR and PP-YOLO models. The main contributions in this thesis comprises of researching and configuring a system through actual implementation of existing models for our polyp application scenario which is capable of detecting and localizing the polyp images in the GI medical images with fairly good accuracy and detection speed. This further training, evaluation and then testing of the model on larger dataset containing both images and videos is highly recommended to for generalization of model. Additionally, the implemented system could also be easily extended to building an automatic diagnosis system with dataset containing multi-class objects of various types of polyps which could perform the medical tasks of detection, localization as well as classification of polyps in GI tract.

The main contributions with regards to the following research question and objectives are as follows

Research question 1. *Can artificial intelligence (AI) techniques such as deep learning in object detection and localization develop a concrete medical diagnosis solution which achieves maximum accuracy and correctness of the outcome in terms of detecting an localizing polyps that has comparable results when manually diagnosed?*

We implemented the recently released improved versions in existing object detection models YOLOv5 [75], YOLOR [130], PP-YOLO [84] using the polyp segmentation Kvasir-SEG [74] and BKAI-IGH NeoSmall-Polyp dataset [29] for evaluating the model performance and detection results. We found that the models showed decent detection accuracy along with good speed of detection for real time.

The thesis work done with regards to the objectives are as follows:
Objective 1: *Implement the training model for object detection and localization in medical images from the gastrointestinal tract.*
The implementation with various configuration settings was done for the three object models with the help of polyp images.

Objective 2: *Validate and evaluate the performance and results of the trained model on GI tract images on unseen medical images and compare it with the ground truth values.*

The models were tested on the unseen test dataset MediaEval-Medico-Polyp [70] and the performance results of the models were obtained.

Objective 3: *Compare the object detection evaluation results of the trained models with the earlier state-of-the-art benchmark models.*
The three models were compared with the earlier benchmark models in the family of YOLO [110] from versins 1 to 4 and ColonSegNet [73] and results were compared and the models showed improved results in FPS.

Objective 4 *The detection model should be expandable for solving other tasks such as polyp segmentation and classification.*

The models can be easily configured for more datasets having different types of polyps and this can be extended to the automation of entire system of polyp detection, localization and classification.

1.6 Ethical considerations

The research work performed in this thesis would be made publicly available for use. The implementation done for the custom dataset is properly referenced from the inspired model. The existing models are utilized to check and evaluate the output for predictions on unseen dataset. The research work in this thesis reports transparency where the findings from this work could be reproduced from the described methods to generate the same results. The medical dataset is difficult to collect and not always available publicly, therefore the research work is carried out with the limited medical dataset available and has scope for improving the prediction accuracy when more and varied datasets become available. To avoid the data bias towards the

colonoscopic images containing only polyps, we have added input data from the colonoscopic examinations that are normal images and do not contain polyps. This is to ensure that the model learns more image features in accordance with the examinations. Another important ethical aspect in medical domain is the privacy and confidentiality of the data which in this work is taken care by manintaining the anonymity of the data from the colonoscopic examinations.

1.7 Thesis outline

The structure for the rest of the thesis is enlisted as follows:

- Chapter 2 - Background: The background chapter describes the medical condition of the gastrointestinal tract in section 2.1, Gastrointestinal Tract in human beings in section 2.2 followed by polyps/lesions in Gastrointestinal Tract described in section 2.3. The section 2.4 explains the CRC its various stages. Further, different screening methods are described in section 2.5. It discusses the need of AI in GI endoscopy field in section 2.6. Next, it lists the various Artificial Intelligence (AI) techniques focusing on object detection models in section 2.7 and section 2.8. In the end it presents the relevant research work with regards to polyp detection and localization task in section 2.9 and ends with a brief summary in section 2.10.

- Chapter 3 - Methodology: This chapter defines the methodology for configuration of the existing models for our custom dataset. It describes the dataset to be used for performing the experiments in section 3.1. Further, introduces the models YOLOv5 [75], YOLOR [130], PP-YOLO [84] briefly in section 3.2. Further the steps required for setting up the environment and configurations for conducting experiments for training all three models are explained. It includes the prerequisites, preparation of the dataset in the model acceptable format, training and validating the model,and performing detection on new image data. It further proceeds with dataset splitting, data augmentation and model training steps followed by different performance metrics to be used. Lastly it presents the summary of the Chapter 3 in section 3.10.

- Chapter 4 - Results: This chapter begins with the presentation of the overall results for all three models along with their corresponding observations. It also highlights the comparison of the results obtained with the earlier models considered as a basis for this thesis.

- Chapter 5 Discussion: This chapter starts with a general discussion and whether the research goals are achieved in section 5.1 and section 5.2. Further, it underlines the challenges and lessons learnt when using deep neural networks for model training in section 5.3 and section 5.4.

- Chapter 6 - Conclusions: The thesis is concluded with short summary in section 6.1, main contributions in section 6.2 and lastly some ideas to continue the work for future scope in section 6.3.

Chapter 2

Background

In this chapter, we present the medical and technical background needed to understand our work and recognize its importance. First, we present some background on the GI tract and discuss some of the most common lesions that appear in this anatomical system. Then, we give an introduction to the machine learning techniques that are used and will be further mentioned throughout the thesis. We further discuss some of the related work around automatic disease detection and localization and finally we address the benefits and shortcomings of related work.

2.1 The CRC medical condition

As discussed in the introduction chapter 1, CRC is the third most common type of cancer in both sexes after breast and lung cancer [69]. The mortality rate is second highest in CRC worldwide [102]. Identifying and detecting any deviations in the digestive system is a difficult process and requires medical expertise. In order to understand the need for developing diagnosis system, we would look through the anatomical aspects of the human digestive system.

2.2 Gastrointestinal Tract

The GI tract consists of all the digestive organs namely mouth, oesophagus, stomach, small and large intestine. The food intake happens through the mouth and the food is digested in order to absorb the required nutrients and energy and the waste is thrown out of the body in the form of faeces. The

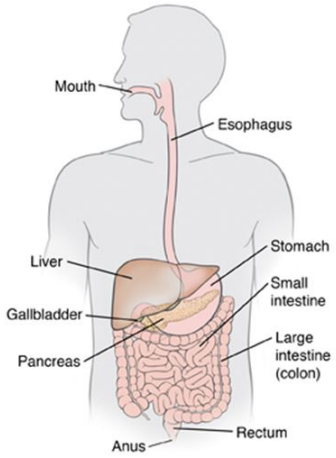

Figure 2.1: Anatomy of Digestive System. Image taken from [1].

anatomy of the GI tract is divided into upper and lower tracts. The upper GI tract is made up of mouth, oesophagus and stomach. The lower GI tract consists of small and large intestine, colon and the rectum [135]. The figure 2.1 [1] shows the overview of the human digestive system. The table 2.1 [13] describes the functions of the different organs in the digestive system briefly.

2.3 Polyps/Lesions in Gastrointestinal tract

Polyps are the lesions which get formed on the innermost lining known as mucosa of the colon as shown in the figure 2.2. These lesions are abnormal changes in the tissues in the colon. Most of the polyps in the colon are not harmful, but few can turn into cancer over the course of time. Therefore, the colon polyps need to be detected and removed earlier to prevent any fatal damage [52].

Colon polyps are categorized into two main groups: neoplastic and non-neoplastic. The non-neoplastic polyps include the hamartomatous polyps, inflammatory polyps and hyperplastic polyps. These non-neoplastic polyps typically do not convert into cancer. Neoplastic polyps consist of the adenomas and serrated polyps [43]. The commonly occurring polyp is adenomatous, and for people who have developed polyps, the chances of it being adenomatous is about in 70% of the cases [101]. Generally, the larger the size of the polyp the more the chances of developing cancer especially with

Digestive Tract Organs	Description
Mouth	This is the first place where the food is moved through the GI tract. The food is chewed and swallowed which passes the food into oesophagus
Oesophagus	Once the food enters oesophagus, the process of peristalsis begins where the muscle called sphincter allows the food to pass into the stomach and prevents flowing back from stomach to oesophagus.
Stomach	The stomach does the mixing of the food with the digestive juices and after one or two hours a thick semi-liquid called chyme is formed. The stomach then empties the chyme produced into the small intestine where further digestion of the food takes place.
Small Intestine	The small intestine mixes the chyme with digestive juices from pancreas, liver and intestine. The absorption of water and nutrients into the bloodstream takes place and the remaining is passed into the large intestine.
Large Intestine	The undigested food, fluid and older cells from GI tract lining form the waste products. The water is absorbed from it and the waste is changed from liquid into stool by the large intestine. The stool is moved into the rectum with the help of peristalsis movement.
Rectum	This is the lower end of the large intestine which stores the stool and pushes it out from the body through the anus when there is a bowel movement

Table 2.1: Functions of different organs in Digestive Tract. Table is taken from [13].

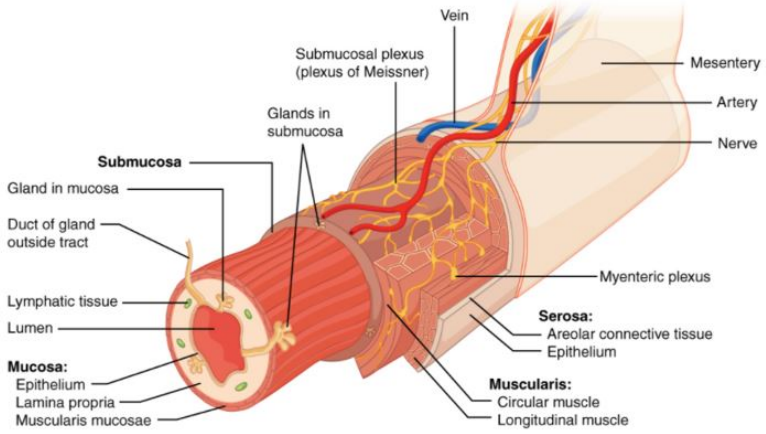

Figure 2.2: Image illustrating Layers of Gastrointestinal Tract. Mucosa is the innermost layer in the colon. Image is taken from [5].

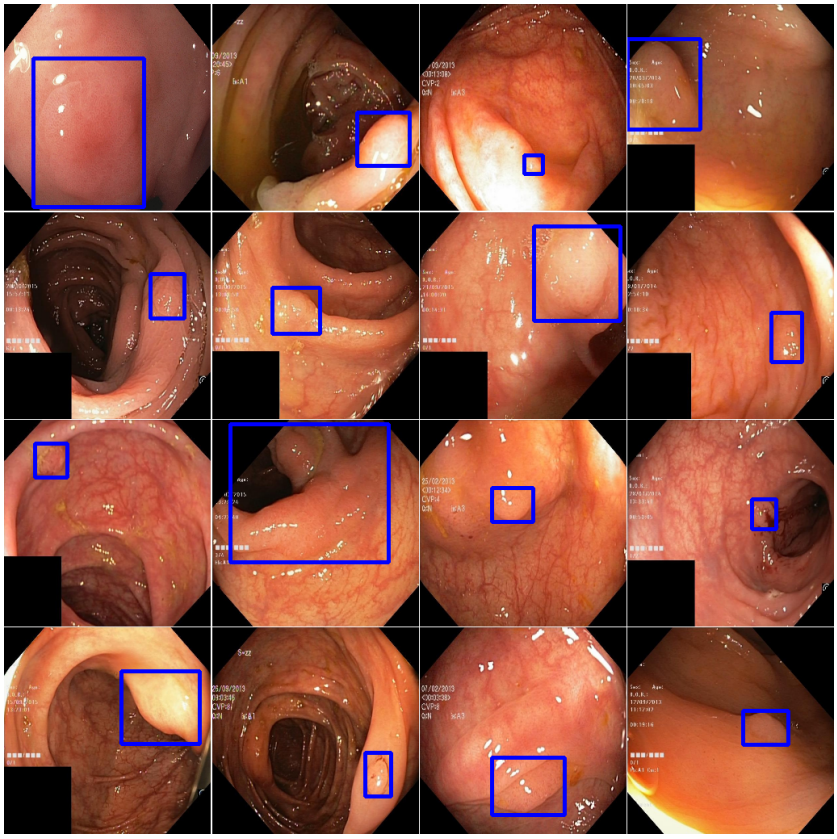

Figure 2.3: Examples of flat and sessile polyps [71]

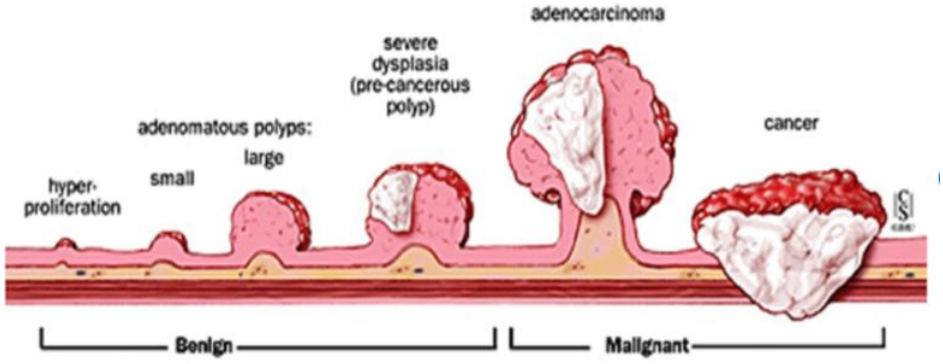

Figure 2.4: Progression from polyps to cancer. Image is taken from [9].

the neoplastic polyps [43].

As shown in the figure 2.4, the polyps if remained in the body for longer periods of time have the potential to turn into cancer [43]. In this thesis, we would implement the system which would assist in identifying all the types of polyps in colon and thereby providing aid to the medical experts.

2.4 Colorectal Cancer (CRC)

CRC is a cancer that originates in the large intestine known as colon or rectum. Initially, it starts with small bunch of non-cancerous tissues known as polyps formed on the inside of the colon. In due course of time, some of the polyps can get bigger in size, undergo cellular transformation and develop into colon cancer. The different screening can help to detect the polyps in time and prevent it from spreading further [4].

CRC staging has similar tumor node metastasis (TNM) classification system that is also used for other types of cancer. There are five stages of CRC from stage 0 to stage 4 [77]. Dukes classification system could also be used for CRC. The table 2.2 provides the details of the various stages of cancer with respect to CRC [14].

CRC is the second most leading cause of cancer mortality globally [38]. CRC is the third most commonly occurring cancer every year in the United States and second most leading cause of deaths in the United States [119]. Norway has got a very high prevalence of colon and rectal cancer. There are nearly 3500 cases of colon or rectal cancer each year in Norway [97]. A researcher and head of the registry division in Norway, Bjørn Møller said the following, *"In Norway, we have had an unfavourable development - colorectal cancer has increased more than in many other countries. While fewer are affected by this cancer in Norway than a few decades ago, we are now at the top in Europe [97]"*.

Adults with age 55 or more are at an increased risk of getting CRC [36]. The broadly classified stages of cancer are localized, regional and distant. Localized stage is where the cancer is local to the organ such as colon or rectum. The regional stage for the cancer is when the cancer has spread to the nearby tissues, organs or lymph nodes in the surrounding region. The cancer stage is termed as distant also known as metastasis if the cancer has spread to the distant tissues or organs in the body [98].

According to the data collected by the National Cancer Institute Surveillance, Epidemiology, and End Results (SEER) program in the United

Colorectal Cancer Stage	Description of the Stages
Stage 0	This is the earliest stage possible and is also called carcinoma in situ. "Carcinoma" refers to cancer that starts in epithelial tissue and "in situ" means original position or place. At stage 0, the cancer is only on the inner side of the colon and has not moved to other place. This stage is also referred as "Carcinoma in situ". The word "carcinoma" means the cancer originates in epithelial tissue and word "in situ" means original location.
Stage 1	At stage 1, CRC has moved from the innermost layer to the middle layers of the colon. This stage is corresponds to Dukes A colorectal cancer.
Stage 2	At stage 2, CRC has moved and extended beyond the middle layers of colon. Sometimes, at this stage cancer has travelled to the nearby organs. This stage corresponds to Dukes B colorectal cancer.
Stage 3	At stage 3, CRC has spread and found in the lymph nodes. This stage corresponds to Dukes C colorectal cancer.
Stage 4	At stage 4, CRC is found in nearby lymph nodes and has spread to the other parts of the body commonly in liver and lungs. It is the most advanced stage of cancer. This stage corresponds to Dukes D colorectal cancer.

Table 2.2: Different Stages in Colorectal Cancer [77, 14, 37].

States for year 2012-2018, the 5 year survival rate for people diagnosed with colorectal cancer overall is 65.1% [98] which indicates that 65% of the people live for at least 5 years after the cancer has been diagnosed. The survival rates vary based on multiple stages in which the cancer is diagnosed as shown in table 2.3.

2.5 Screening Methods

GI tract could be vulnerable to several diseases, infections, inflammations or cancers. The prognosis of GI tract diseases could be improved with early screening and appropriate treatment to reduce the mortality rates due to

Cancer Stage	**Percentage Diagnosed**	**Five Year Survival Rate**
Localized	37%	90.9%
Regional	36%	72.8%
Distant	22%	15.1%
Unstaged	5%	40.5%
Overall	100%	65.1%

Table 2.3: CRC stat facts for year 2012-2018 from National Cancer Institute Surveillance, Epidemiology and End Results (SEER) Program. Table is taken from [98].

problems in GI tract [40]. Different screening methods are explained and the difficulties associated with them are discussed.

According to the Statistics adapted from the American Cancer Society's (ACS) publication named Cancer Facts & Figures 2020 [22], death rate in 2017 was 54% less when compared to what it was in 1970 due to CRC. This is mainly because of the improved treatments and rise in the screening methods in order to find any abnormal changes in the colon and treat it at earlier stages before turning into cancer [44].

For preventing diseases of GI tract and timely treatment of any possible medical condition, the American Cancer Society 2018 guideline recommends to undergo regular colorectal screenings for people above 45 years of age [23]. Another problem with the CRC is that the GI tract does not show any visible symptoms of any medical conditions until it has moved to an advanced stage where the condition becomes serious. Therefore it is necessary that GI tract is screened timely to prevent any disease or development of polyps. Below are the list of the various current state of art screening methods for GI tract which would be described in detail as below.

2.5.1 Traditional Endoscopy

Endoscopy as the name indicates is looking inside, is a medicinal procedure to look through the inside of the body [**?**] [134]. According to Martin Culjat et al. , endoscopy is a *"small telescope device(s) to look inside the body that applies generally to the optical devices (telescopes) used for endoscopic procedures" [89].* This definition is perhaps more suitable in a non-technical setting. Another definition described by Kay Ball is as follows, "*The inspection of body organs or cavities by means of an endoscope, which is a device consisting of a tube and optical system* [26]". This description explains what endoscopy means and is more relevant in medicinal field.The table 2.4 describes some of the significant milestones achieved in the history of evolution of endoscopy adapted from [18].

Year	**Important Persons or entity**	**Contribution**
1806	Phillip Bozzini	The "Lichleiter", candle illuminated in a container reflecting light by an angled mirror.
1853	Antonin Desmoreaux	First used the term "endoscopy", used kerosene lamp with paraffin flame and 45 degree angled mirror.
1879	Max Nitze	Created cystoscope with water cooled platinum filament lamp and series of lenses in a metal tube
1879	Thomas Edison	Invention of incandescent light bulb
1883	David Newman	Replaced platinum wires with incandescent light bulb on endoscope
1910	Hans Christian Jacobaeus	First published laproscopy in a patient
1948	Harold Hopkins	Developed zoom lens
1956	Basil Hirschowitz and Larry Curtis	Replaced Hopkin's glass fibers with better flexible optical fiber material and glass coating
1959	Harold Hopkins	Developed rod-lens system
1966	Harold Hopkins and Karl Storz	Designed new rigid endoscope with self-focusing lens
1971	Hiromi Shinya and William Wolff	First polypectomy during colonoscopy
1985	Erich Muhe	First laproscopic cholecsytectomy
1993	Becker et al.	First report of 3D endoscopic system
2000	FDA	First systems for general robotic surgery approved for use in humans

Table 2.4: History of evolution of endoscopy. Table is taken from [18]

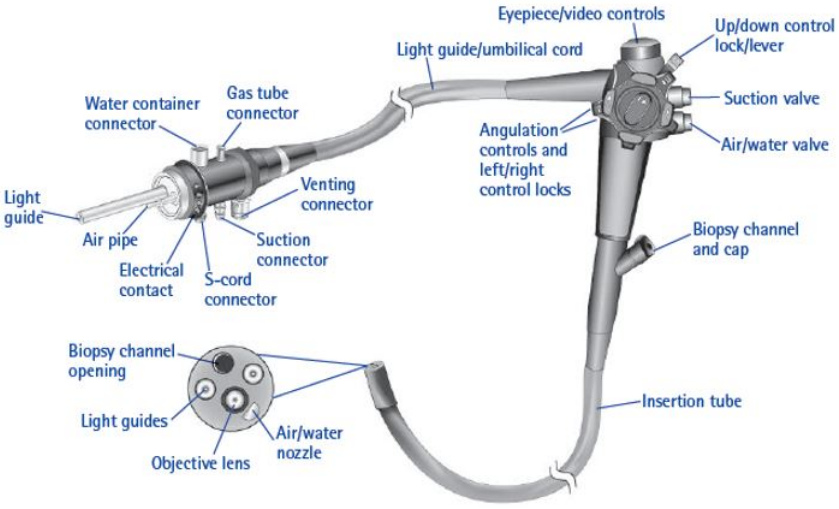

Figure 2.5: Flexible Endoscope showing different parts. Image is taken from [10]

An instrument known as endoscope is used to examine the inner side of a cavity of the body. The endoscope has a flexible tube with a light and a camera attached to it through which the doctors can see the pictures of the organs being examined on the color TV monitor [131]. The figure 2.5 shows the different parts of the flexible endoscopes.An endoscopy can be used to diagnose unusual health symptoms [100]. There are mainly two types of endoscopy of the gastrointestinal tract : upper endoscopy and colonoscopy. Upper endoscopy is the procedure where the endoscope is inserted through the mouth to view esophagus, stomach, and small intestines. Colonoscopy is the procedure where the endoscope is inserted through the rectum to see the lining of the large intestine, colon and rectum [68].

2.5.2 Upper Endoscopy

An upper endoscopy procedure is to investigate the upper digestive system visually through a tiny camera which is attached on the end of a long and flexible tube. A specialist person also known as gastroenterologist performs the examination to diagnose or treat medical conditions affecting esophagus, stomach and duodenum (beginning of the small intestine). The upper endoscopy in medical terms is known as esophagogastroduodenoscopy. The upper endoscopy could be performed in doctor's office, an outpatient surgery

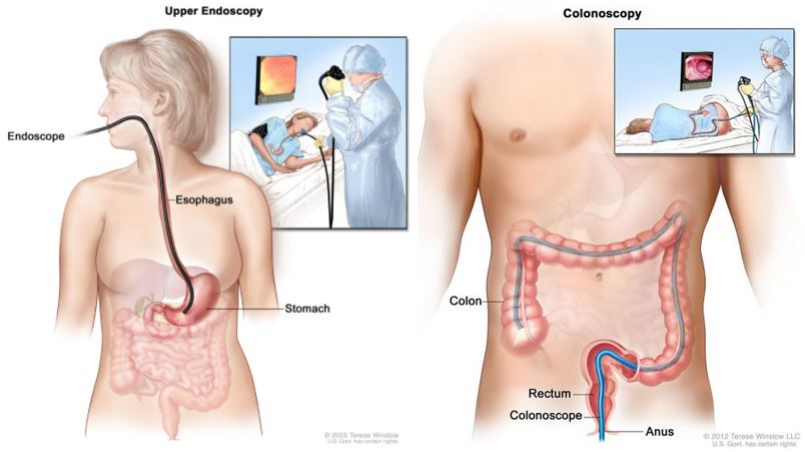

Figure 2.6: Upper and Lower Endoscopy. Image is taken from [6]

center or in a hospital. During an upper endoscopy procedure, monitors are attached to the body to monitor the breathing, blood pressure and heart rate. The patient may receive sedatives or may receive a spray of an anesthetic in the mouth. Then the endoscope is inserted in the mouth and doctor asks to swallow as the scope passes through the throat. While the endoscope passes through the esophagus, a tiny camera sends the images to the video monitor in the examination room. Sometimes special surgical tools are passed to collect tissue sample or to remove polyps [42].

2.5.3 Colonoscopy

Colonoscopy procedure is used to examine the lower gastrointestinal (GI) tract. Colonoscopy is used to detect any changes or medical abnormalities in the large intestine (colon) and rectum. For colonoscopy, a sedation or an anti-anxiety medicine is given. A long flexible tube (known as colonoscope) is inserted into rectum. A tiny camera displays the images of the colon on the video screen. Colonoscopy is used to detect intestinal signs and symptoms in case of medical problems. It is also used for screening of colon cancer and to detect any polyps. Other treatments including taking tissue samples (biopsies) and polyp removal may also be performed [91] [114].

2.5.4 Flexible Sigmoidscopy

Sigmoidscopy is an examination to investigate the lower part of the large intestine (colon). During the procedure a thin flexible tube known as sigmoidscope is inserted into the rectum. In this procedure, the sedation is generally not required. A small camera attached to the tip of the tube allows the doctor to examine the inside of the rectum, the sigmoid colon and descending colon which is the last 2 feet (around 50 cm) of the large intestine. The entire colon cannot be viewed through the flexible sigmoidscopy. Therefore, this exam cannot alone detect cancer or any polyps that could have developed in the colon [41].

2.5.5 Virtual Colonoscopy

Virtual colonoscopy is an examination that is minimally invasive in order to screen for cancer of the large intestine also known as colon cancer. In traditional colonoscopy, a scope is inserted into the rectum and moved through the colon to analyse the colon. However, in virtual colonoscopy, a CT scan is done to generate multiple cross sectional images of abdominal changes. These images are then combined and digitally manipulated to give a detailed view of the colon and the rectum. Virtual colonoscopy is only one of the options for screening colon cancer [24].

As virtual colonoscopy does not need the use of a colonoscope, therefore this type of examination is more preferable for some people. There is no need to take sedation and hence the person can get back to his normal routine activities or go home once the exam is completed. The absence of sedation has lower risk for the people who could have harmful reactions to the sedative medications given in the traditional colonoscopy. This procedure is also less time consuming when compared to the conventional colonoscopy. Virtual colonoscopy also has additional benefit of detecting any diseases or abnormalities outside of the colon [136].

2.5.6 Wireless capsule endoscopy

This procedure involves a wireless camera which is tiny so that it could easily fit into a vitamin sized capsule that can be swallowed. Once the capsule is

swallowed, it travels through the digestive tract and thereby takes thousands of images which are sent to the device connected to the wearable belt. One advantage of the capsule endoscopy is that the small intestine can be examined which is on the other hand difficult using the standard colonoscopy. The capsule swallowed is out of the body through excretion process within 24-48 hrs. Therefore, this type of endoscopy is relatively safe method for diagnosis [99].

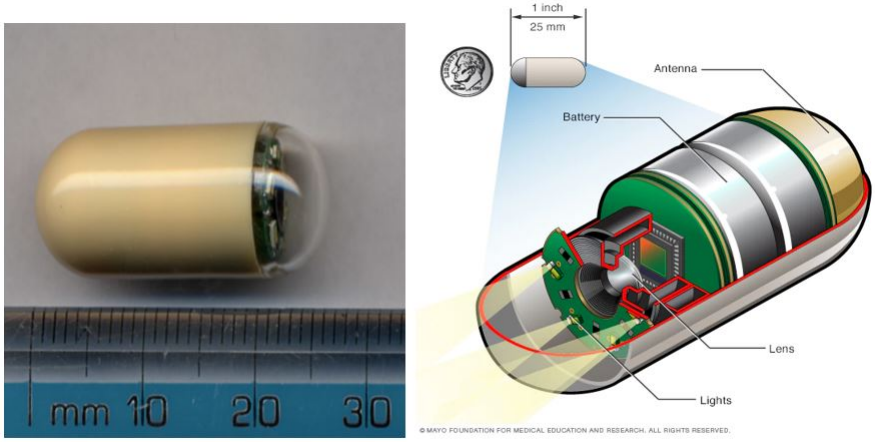

Figure 2.7: Wireless Video Capsule Endoscopy. Image is taken from [3]

2.6 Need of AI in the GI endoscopy

The GI endoscopy is golden standard for diagnosing and treatment of diseases occurring in the GI tract. Around 2.8 million cancers in GI tract including esophageal, stomach and colorectal are detected every year across the globe. Several among these cases could be prevented with the help of improved performance in the endoscopy and ensuring high quality systematic screening [32]. These type of cancers pose a significant health challenge to the society due to higher mortality rate of 65% [12] and CRC specifically is the second most cause of deaths due to cancer globally [39].

The quality of endoscopes has improved in the last 10 to 15 years. However, the correct results from an endoscopic examination relies highly on the endoscopist operator skill set, level of expertise, work attitude, knowledge and temperament [61]. The other reasons contributing to the missed

colorectal polyps during examination include: insufficient preparation of the colon, peristaltic movement, anatomical structure of the colon, use of old instruments and monitors,stress,human errors by physician [96]. The average polyp miss rate of 20% occurs in the colon [50].

Due to health risks of CRC, colon polyp detection using AI has great potential in the medical research field. Moreover, a missed colorectal polyp could potentially result in CRC development [43]. Use of AI techniques can significantly contribute to the detection and diagnosis of polyps.

2.7 Artificial Intelligence Techniques

AI has brought technological revolution in all the fields. A major and key area for the applications of AI techniques is largely in the fields of medicine. AI has been researched and studied especially in the field of gastroenterology with its diverse applications such as assistance in risk detection, diagnosis and pathological identification. It is a hot topic of interest in the endoscopy and has immense potential to transform the gastroenterological practices. AI has revolutionalized all the aspects of modern endoscopy ranging from the cancer screening to the automated report generation [19].

AI in simpler terms is referred to as intelligence that is displayed by machines, as against the intelligence which is naturally shown by animals including human beings [132]. The figure 2.8 shows hierarchy structure of the AI domains. Machine Learning (ML) is a sub-field of AI where the training algorithms learn from the different patterns in the dataset instead of programming [103] [57]. Representation Learning (RL) is a sub-field of ML where learning for the training algorithms happens on the best features which is utilized for classification of data [103] [57]. Deep Learning (DL) is a type of RL where the combination of features are learnt that represent different hierarchical structures in the data that output image classification results [103] [57].

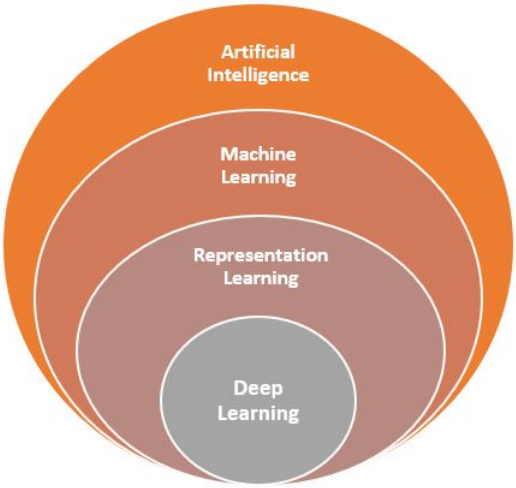

Figure 2.8: Hierarchy of artificial intelligence Domains

2.7.1 Deep learning approaches

Deep learning algorithms are subset class of the machine learning algorithms. The definition of Deep Learning in article [51] is described as : Deep learning contains multiple layers for extracting the higher- level and key features from the input training data. For instance, in case of image processing, the lower layers could identify edges and the higher layers could determine the human related concepts namely digits, letters or faces [51] [133]. The figure 2.9 shows the architecture of a deep learning network containing several layers for feature extraction and classification to learn and extract features from the input dataset.

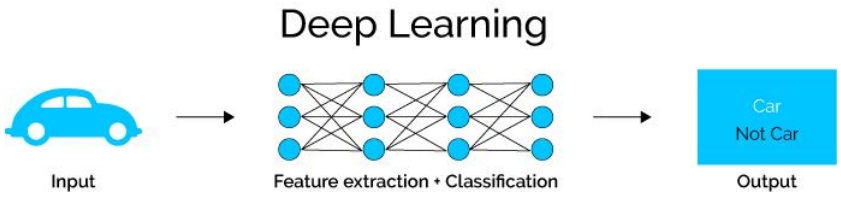

Figure 2.9: Deep Learning Architecture. Image is taken from [2]

2.7.2 Convolutional Neural Network

Convolutional neural network (CNN) is a neural network class that specializes in visual image analysis such as image recognition and classification. When we look at an image, human brain is processing a large amount of information.

The entire visual field is covered as each neuron works in its receptive field and connected to other neurons. Similar to human brain, each neuron in CNN only processes data in its receptive field. The layers are organized such that they detect simple patterns such as lines, curves, etc first and later complex patterns such as faces, objects, etc [94].

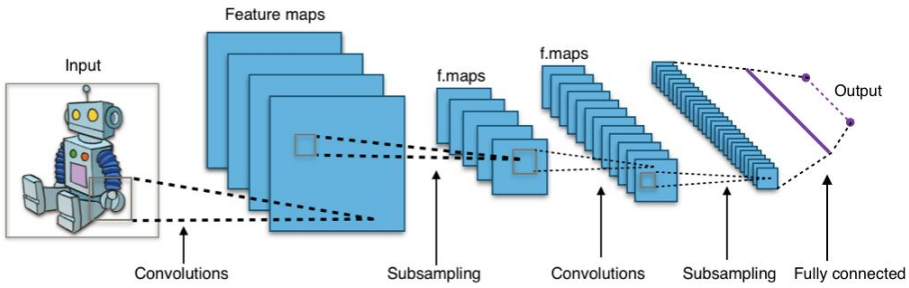

Figure 2.10: Typical Architecture of Convolutional Neural Network. Image is taken from [46]

A CNN is made up of (1) convolutional and pooling layers (2) fully connected layers. The convolutional and pooling layers are main components for extracting different features, and fully connected layers to perform classification. Multiple feature maps are created in the process of applying various filters to the input image for extract key features. This pre-processing step of filtering is known as convolution. For making a CNN model successful, the feature maps are compressed by pooling pixels to smaller sizes in order to capture a larger section of the image. The convolutional and pooling layers are iterated multiple times. The fully connected layers generate the final outcomes by combining all the features [143].

2.7.3 Computer Vision

Computer vision CV is defined as an interdisciplinary field of study that aims to help computers visualize, analyze and understand the contents of the images and videos. This problem appears to be simple since people can trivially solve this problem with ease. However, it still remains an unsolved problem mainly due to complexity in visual perception dynamically and highly changing physical world [7]. Computer Vision therefore is a field of study focusing with the goal of helping the computers to see. It can be broadly classified as a sub-field of artificial intelligence and machine

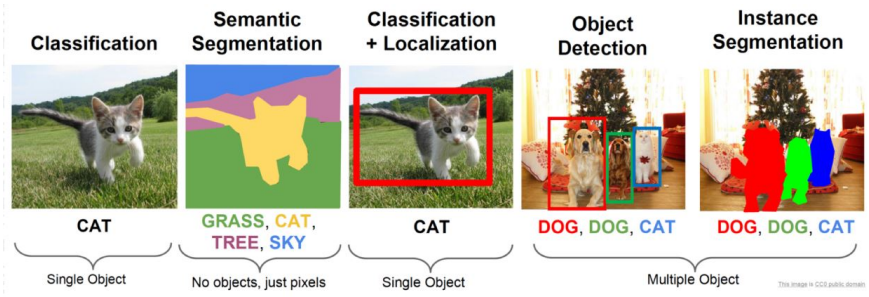

Figure 2.11: Computer vision tasks. Image is taken from [49]

learning which uses specialized methods and use general learning algorithms. The popular applications of the computer vision include trying to recognize certain things in the images [7].

Object recognition involves tasks in identifying objects in the images. Image classification aims at predicting the class of the object found in the image. Object localization focuses on identifying the location of one or more objects identified in the image and drawing a bounding box around their extent. Object detection is a combination of these two tasks that localizes and classifies one or more objects in the image. Below is a brief explanation in terms of input and output of computer vision tasks [7].

- **Image Classification**: Predict the type or class of an object in an image. Input is an image with a single object. Output is a class label specifying the classification type of the image [7].

- **Object Localization**: Locate the presence of objects in an image and indicate their location with a bounding box. Input is an image with one or more objects. Output is one or more bounding boxes for specifying the object location in the image [7].

- **Object Detection**: Locate the presence of objects with a bounding box and types or classes of the located objects in an image. Input is an image with one or more objects. Output is one or more bounding boxes with a class label assigned for each bounding box to specify the location and class of the detected objects [7].

- **Object Segmentation**: Highlight the specific pixels belonging to the located objects in an image. Input is an image with objects. Output is highlighting bounded boxes for objects in the image [7].

The main focus in this thesis is on the task of object detection and localization of polyps in a given image from the endoscopic examinations. With regards to this, the next section describes different available state-of-the-art models for object detection and localization tasks.

2.8 Object Detection Models

As discussed earlier, object detection deals with identification and labelling of the objects in an image, videos and even real time footage. The object detection models need to be trained with huge amount of annotated visual data which in turn helps to process the information in the new data. The object detection has a key component which is a bounding box represented by either a square or a rectangle which identifies the edges if the object detected in the image or video. In addition, the bounding box is also tagged with a label of the object viz. a cycle, a person, a dog for describing the target object. There can be overlap of bounding boxes for highlighting multiple objects in a given image [123].

A object detector normally has two components: a backbone and a head. A backbone is pre-trained on ImageNet and a head is used to predict the classes and the bounding boxes of the found objects [123]. The backbone of the detectors running on GPU platform could be VGG [118], ResNet [59], ResNeXt [139],or DenseNet [65] while those running on CPU platform, the backbone could be SqueezeNet [67], MobileNet [64] [115] [63] [126] or ShuffleNet [145] [86].The head has two types: one-stage object detector and two-stage object detector. Common two-stage object detectors are R-CNN [56] series,fast R-CNN [55], faster R-CNN [111], R-FCN [47].The common representative models for one stage detector are YOLO [106] [109] [110], SSD [83], and RetinaNet [80].

There are mainly two approaches for object detection models namely: deep learning and machine learning. The deep learning approaches are widely regarded as the state-of-the-art approach as it is intuitive and requires less human intervention. Object detection with deep learning employs the concept of convolutional neural networks CNN. These neural networks represent the same complex neurology of the human brain. The neural network learn-

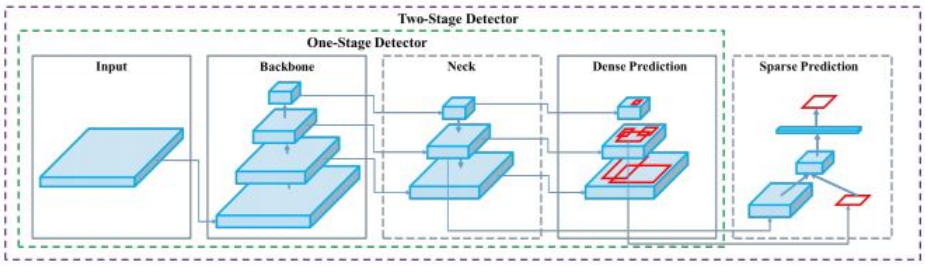

Figure 2.12: Typical Structure of Object Detector. Image is taken from [30]

ing could be supervised, semi-supervised or unsupervised depending on the amount of annotated training data. Deep neural networks provide the fastest and most accurate results for single and multiple object detection as they have ability of automated learning with less manual engineering. The object detection models are pre-trained with datasets like COCO (Common Objects in Context) that contain thousands of visuals that gives headstart to to configure models using custom dataset [123].

Detection methods are useful to predict the class of the object in the image and the localization methods produce the bounding boxes around the detected object. The object detection algorithms can be classified into two groups:Classification-based and Regression-based algorithms [93].

- *Classification-based algorithms* [93]: It consists of two stages. In the first stage, it selects a bunch of Region of Interest (ROI) in the image where it has high probability of locating the objects. In the second stage, it applies a Convolution Neural Network in the selected regions for detecting presence of an object. A problem with this algorithm is to execute the detector for every ROI which makes the algorithm slow and and costly. The R-CNN [56], Fast R-CNN [55] and Faster R-CNN [111] models belong to this category.

- *Regression-based algorithms* [93]: In this method, no ROI is selected in the image, but instead it predicts the classes and bounding boxes for whole image at once making detection faster than classification algorithms. YOLO [106] [110] detector also known as ("You Look Only Once") belongs to the regression based algorithm and is very fast and is also used in the real time object detection.

The following sections describe some of the prominent object detection models that fall into category of region based convolutional neural network such as R-CNN [56], Fast R-CNN [55] and Faster R-CNN [111]. Further, the modern object detection YOLO [106] [110] family of models would be discussed.

2.8.1 R-CNN

The R-CNN model was described by Ross Girshich, et al. in paper [56] in 2014. The model approach based on convolutional neural network showed the state-of-the-art results for object detection on the VOC-2012 dataset [54] and 200-class ILSVRC-2013 dataset [113]. Figure 2.13 explains the architectural overview of the R-CNN model. The proposed R-CNN consists of the following three modules [34]:

- **Module 1: Region Proposal** To generate and extract region proposals which are independent of category. For instance, candidate bounding boxes.
- **Module 2: Feature Extractor** To extract feature from each candidate region by using a deep convolutional neural network.
- **Module 3: Classifier** To classify features into one of the known classes.

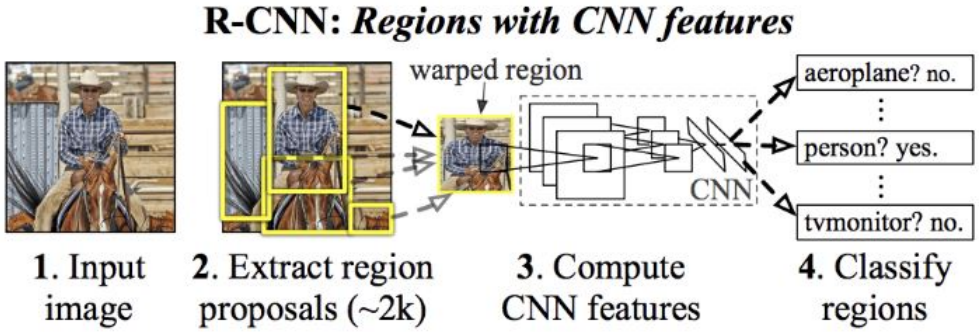

Figure 2.13: R-CNN object detection system overview. Image is taken from [56].

A computer vision algorithm called "selective search" proposes the candidate regions or bounding boxes for detecting the interested objects

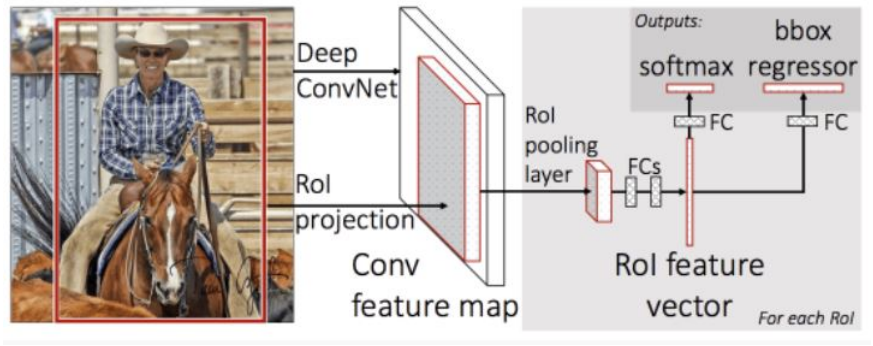

Figure 2.14: Fast R-CNN model architecture. Image is taken from [55].

in the image. This model also enables the flexibility to utilize different region proposal algorithm. Although the R-CNN [56] approach is simple application of CNN for solving the object localization and detection it has some disadvantages. The training using R-CNN [56] model is multi-stage pipeline which is slow and expensive in space and time as it requires a CNN-based feature extraction for every candidate region generated by region proposal algorithm. Additionally, the object detection with R-CNN has less speed [34].

2.8.2 Fast R-CNN

The problems with the speed of R-CNN was addressed by proposing the model Fast R-CNN in 2015 research paper [55]. Researchers proposed a single model rather than a pipeline for learning and extracting regions and then later directly classifying. The figure 2.14 explains the architecture of the model. It takes the input as an image, a set of region proposals that passes through the deep CNN. For feature extraction, a pre-trained CNN is used. The last layer in the network is a custom layer known as Region of Interest Pooling Layer or ROI pooling to extract the features particular to the given candidate region. The output from the CNN is analyzed by a fully connected layer that divides the output into two parts, one via a softmax layer for class predictions and another one with a linear output for bounding box. The process iterates for several times for every region of interest in the input image. The training of the model and making predictions is quite fast. However, the model needs proposed set of candidate regions for each input image [55] [34].

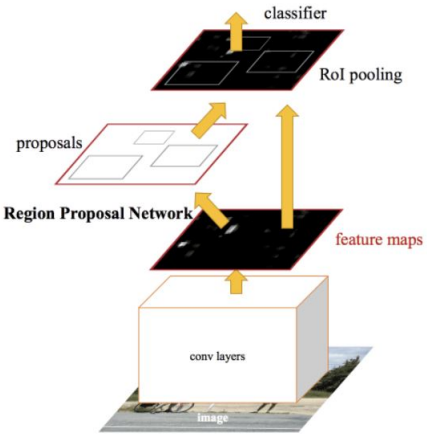

Figure 2.15: Faster R-CNN model architecture. Image is taken from [112]

2.8.3 Faster R-CNN

An improvement to the architecture of the model for Fast R-CNN [55] in terms of speed of the training and detection was proposed as Faster R-CNN in 2016 research paper [112]. The aim of the architecture was to refine the region proposals for training called as Region Proposal Network (RPN) that are utilized in a single model design into Fast R-CNN [55] model. This in turn helps in reduction of number of region proposals and to accelerate the test-time of the model close to real-time with the performance of the existing state-of-the-art models.

The model architecture is based on a single unified model and contains two modules [112] [34]:

- **Module 1: Region Proposal Network** Consists of CNN to propose the regions and the type of object to be considered in the region.

- **Module 2: Fast R-CNN** This consists of CNN to extract the features from the proposed regions and to give the output for bounding box and class labels.

The figure 2.15 shows the architecture of the Faster R-CNN model [112]. The RPN serves as an attention mechanism to the Fast R-CNN network for providing the information on where to look or give attention. The RPN takes the output from pre-trained deep CNN and then sends through a small network over feature map with output of several region proposals and having corresponding class prediction. The region proposals from the output of RPN

are the bounding boxes which are anchor boxes or having pre-defined shapes for enhancing and improving the proposal of regions. The predictions on the class are binary for determining the presence or absence of the object also called “objectness” of the proposed region [112] [34].

2.8.4 YOLO

YOLO [107] also known as ("You Look Only Once") is an object detection model based on the Deep Learning based approach. The YOLO [107] detector belongs to the regression based algorithm. The algorithm separates the bounding boxes and associated class probabilities. Therefore, this network can be optimized on the detection performance. The unified detection technique used by YOLO[107] segregates various components of the object detection into a single feed neural network [87].

The figure 2.16 shows the methodology for YOLO [107] model. The algorithm of YOLO [107] model takes the input image and divides it into several grids and for each grid it calculates the probability value for an object to be present in that grid. In the next step, it combines and groups the surrounding grids with high value probabilities into a single object. In similar way, the model training is performed where the center of the detected object is compared against the ground truth and the model weights are adjusted accordingly [107] [87]. YOLO[107] achieved 63.4 mAP (mean average precision) and having an inference speed of 45 FPS (frames per second)that was significantly higher than the other prevalent object detection state-of-the-art methods at that time [107]. The original YOLO [107] (You Only Look Once) was written in custom framework called Darknet [105] which is written in low level languages. The YOLO family has evolved over time since initial release in 2016 and produced the following versions of YOLO family for realtime object detectors in computer vision: YOLOv2 [108], YOLOv3 [110], YOLOv4 [30], YOLOv5 [75] and PP-YOLO [84].

In the research paper [121], the authors through various comparisons for different detection algorithms concluded that although Faster-RCNN [112] and SSD [83] yield good accuracy but for achieving higher speed in detection for real time, YOLO [107] is the best model. Also, in the study in paper [122],

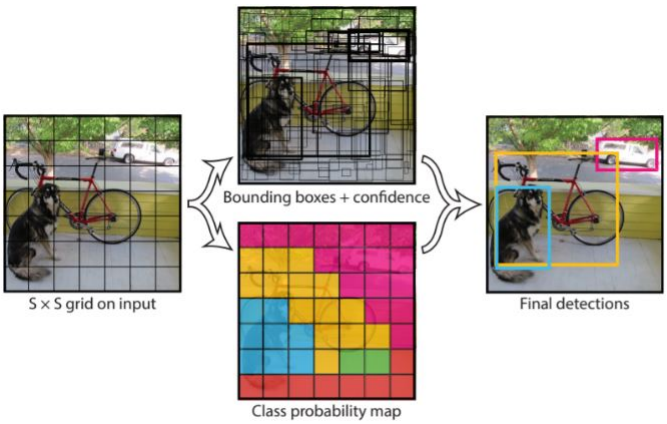

Figure 2.16: YOLO model methodology. Image is taken from [107]

the results suggest that Mask-RCNN [58] requires more time for detection than YOLO [107]. Then the YOLOv2 [108] was introduced which is able to detect nearly more than 9000 categories of different object. Again, in YOLOv3 [110] was released with better features and having speed three times faster than SSD [83]. YOLOv4 [31] is the next improved version on YOLOv3 [110], that achieves 10% higher Average Precision and 12% better FPS speed. Similarly, the newer version YOLOv5 [75] have showed improved accuracy over YOLOv3 [110] and YOLOv4 [31] but the speed is similar to YOLOv4 [31]. Another state-of-the-art machine learning algorithm for object detection is YOLOR [130] which stands for You Only Learn One Representation is different from YOLO versions 1 to 4. The research paper proposed YOLOR as a "*unified network to encode implicit knowledge and explicit knowledge together*" and states that the results of YOLOR model show advantage from using the implicit knowledge [130]. The table 2.5 shows the comparison of several deep learning based object detection models in terms of mean average precision (mAP) and frames per second (FPS).

In this thesis work, we would utilize YOLOv5 [75], YOLOR [130] and PP-YOLO [84] object detection models for doing configurations for our custom polyp dataset for our application scenario of polyp detection and localization as these models are released recently and have demonstrated good performance over the existing state-of-the-art models. Also, another objective is to test the performance of these models specific to the polyp detection and compare the results with the existing benchmark models. The architectures for these models would be discussed in detail in chapter 3.

Detection Model	Mean average precision (mAP) in %	Frames per second (FPS)	Test Dataset
Fast R-CNN [55]	70.0	0.5	Pascal-VOC 2007
Faster R-CNN VGG-16 [111]	73.2	7	Pascal-VOC 2007
Faster R-CNN ResNet [111]	76.4	5	Pascal-VOC 2007
YOLO [107]	63.4	45	Pascal-VOC 2007
YOLOv2(288×288) [108]	69.0	91	Pascal-VOC 2007
YOLOv2(352×352) [108]	73.7	81	Pascal-VOC 2007
YOLOv2(416×416) [108]	76.8	67	Pascal-VOC 2007
YOLOv2(480×480) [108]	77.8	59	Pascal-VOC 2007
YOLOv2(544×544) [108]	78.6	40	Pascal-VOC 2007
YOLOv3(320×320) [110]	28.2	22	MS-COCO
YOLOv3(416×416) [110]	31.0	29	MS-COCO
YOLOv3(608×608) [110]	33.0	51	MS-COCO
YOLOv4(512×512) [31]	43	83	MS-COCO
YOLOv4(608×608) [31]	43.5	65	MS-COCO
YOLOv5s [75]	69.9	111	MS-COCO
YOLOR [130]	73.3	72	MS-COCO
PP-YOLO [84]	45.2	68.9	MS-COCO

Table 2.5: Comparison of various deep learning based object detection frameworks.

2.9 Related work

A lot of research work has been done in the medical area of automated polyp detection in the past. In the recent years, CNNs have gained a lot of popularity [124], [116] and has been used extensively in the public challenges [28], [21].

The table 2.6 show that the research studies have achieved decent performance for object detection and localization. The study [148] have shown better results with color and texture features using the CNN based trainable feature extractor. The researchers in the paper [144] have proposed offline and online three dimensional framework using 3D CNN. The study[147] aims at detection and classification of hyperplastic and adenomatous colorectal polyps through transfer learning using deep convolutional network. In the paper [95] have proposed a Y-Net deep learning method which surpassed the earlier state-of-the-art methods with 7.3% F1-score and 13% recall improvement. The paper[117] addresses the polyp detection in colonoscopy videos by exploiting the bi-directional temporal dependencies in the sequential im-

Research Study	Year	Localization Type	Multiple polyp	Real time	Sensitivity	Specificity	Precision	FPS
Zhu R. et al. [148]	2015	Bounding box (16x16 patches)	Yes	No	0.7944	0.90	0.63	-
Yu et al. [144]	2017	Bounding box	No	No	0.71	-	0.881	-
Zhang R. et al. [147]	2018	Bounding Box	No	No	0.7160	-	0.8860	6.5
Mohammed et al. [95]	2018	Bounding Box	No	Yes	0.8440	-	0.8740	-
Qadir et al. [117]	2018	Bounding Box	No	Yes	0.8030	0.8650	-	-
Liu et al. [82]	2019	Bounding Box	No	Yes	0.8030	-	0.7360	32
Zhang et al. [146]	2019	Bounding Box	No	Yes	0.7637	-	0.9392	50
Lee et al. [79]	2020	Bounding Box	No	Yes	0.9670	-	-	67.17

Table 2.6: Research studies in polyp detection and localization.

age frames in a video using CNN. The researchers in the paper [82] have investigated the use of single shot detector (SSD) [83] framework to detect polyps in the colonoscopy videos. In another study [146], a CNN based on Single Shot MultiBox Detector (SSD) [83] architecture is developed achieving polyp detection speed of 50 frames per second (FPS) with an improved mAP of 90.4%. The research study [79] has utilized YOLOv2[108] for algorithm design for automatic polyp detection and achieved a speed of 67.16 frames per second.

The research works have been substantial in the medical field of polyp detection and localization. However, there is a need to achieve good detection speed in terms of frames per second for faster processing and to improve the real time detection speed. The polyp localization in a colonoscopic image or video frame should be as accurate as possible to particularly find the polyp location with maximum precision value and to reduce the polyp miss rates and false negatives cases. Therefore, there is a need for research in this area.

2.10 Summary

In this chapter, the relevant background and related work carried out in the past were discussed. The human digestive system and particularly

gastrointestinal tract was described briefly. In addition, we looked through different types of endoscopy procedures and various types of screening methods including traditional endoscopy, colonoscopy, virtual sigmoidscopy and video capsule endoscopy. The traditional endoscopy procedures are time consuming and can sometimes also be uncomfortable for patient. However, the video capsule endoscopy screening method is performed with a capsule with tiny camera is swallowed by the patient and the images and videos are captured, transmitted and stored at the receiver end for later analysis and diagnosis. We further discussed the several AI techniques such as deep learning, convolutional neural network and computer vision. Further, the popular object detection models were explained and the various past research works in this area.

To conclude, the earlier research studies were carried for polyp detection and localization using CNN and deep learning based networks using image and video datasets. The earlier works have shown good potential in achieving high accuracy and precision values. But most of the studies miss the frames per second evaluations. This leads to an opportunity to develop an automated polyp detection and localization model with higher precision and detection speed for aiding the medical domain in real time.

Chapter 3

Methodology

This chapter explains the methodology used to implement the models and the configurations required for polyp detection and localization task. It describes the dataset to be used for the configuration and experiments and the architecture overview of different already existing object detection models used, different parameter settings and configurations for experimental setup for model training and testing and description on the performance metrics used for evaluating the model performance.

3.1 Dataset

This section presents the dataset which would be utilized for the task of polyp detection. The Kvasir-SEG [74] dataset obtained from the various colonoscopic examinations and dataset BKAI-IGH NeoPolyp-Small [29] would be used to develop object detection and localization model in this thesis.

- **Kvasir-SEG [74]:** This dataset is a collection of gastrointestinal polyp images and their corresponding segmentation masks that are annotated and verified manually by gastroenterologists. The dataset consists of 1000 polyp images along with their corresponding ground truth. The images in the dataset have varied resolution from 332×487 to 1920×1072 pixels. The images and their segmentation masks are available in two different folders having same filename. The images are encoded with JPEG compression. The JSON file contains the details for the bounding box (coordinate points) in the corresponding images.

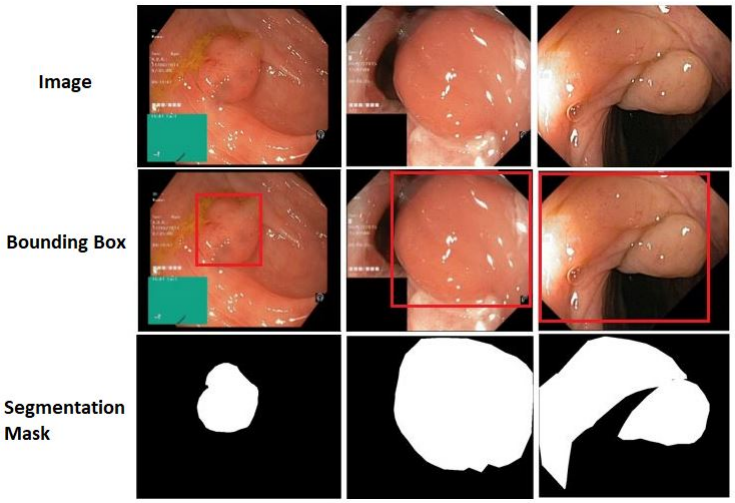

Figure 3.1: Colonoscopic polyp, corresponding bounding boxes and segmentation masks from Kvasir-SEG [74]

The figure 3.1 shows the images from the colonoscopic examinations taken from the dataset Kvasir-SEG [74] along with their corresponding from bounding boxes in red colour and segmentation masks.

- **BKAI-IGH NeoPolyp-Small** [29]: This dataset is publicly available and has 1200 images. This dataset contains 1200 images with 1000 images in training set and 200 images in test set. The polyps are classified into neo-plastic and non-neoplastic denoted with red and green colour respectively. The dataset can be utilized for polyp segmentation and polyp detection and localization with identification of polyp neoplasm characteristics. The figure 3.2 shows the images from the colonoscopic examinations taken from the dataset BKAI-IGH NeoPolyp-Small [29] with neoplasm characteristics with their bounding boxes in red colour and segmentation masks.

3.2 Model selection

The benchmark models for the task of object detection were discussed earlier in the Section 2.8. The recent research studies published the newer and improved versions of YOLO [107] architecture model called YOLOv5 [75], YOLOR [130] and PP-YOLO [84]. These models showcased improved performance when implemented against COCO dataset [81]. As seen in

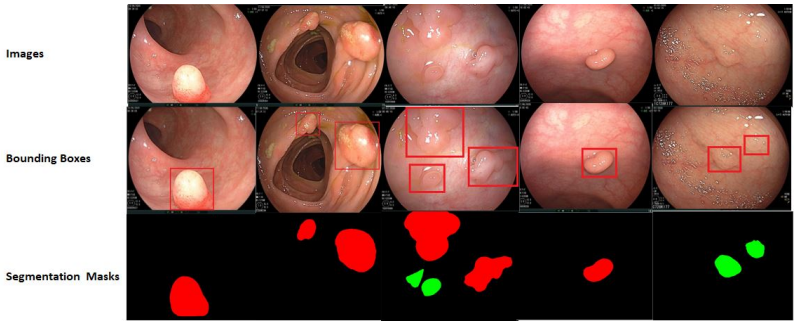

Figure 3.2: Colonoscopic polyp, corresponding bounding boxes and segmentation masks from BKAI-IGH NeoPolyp-Small [29]

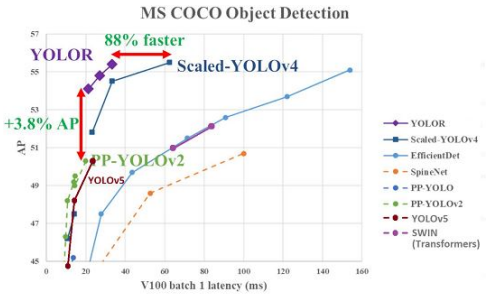

Figure 3.3: Comparison of YOLOR [130] on MS COCO dataset with various models. Image is taken from [76]

the figure 3.3, the YOLOR [130] model demonstrates 3.8% higher average precision value than PP-YOLOv2 [66] and 88% higher FPS speed than Scaled-YOLOv4 [128]. Similarly, the figure 3.4 shows that PP-YOLO [84] executes faster than the YOLOv4[31] and the mAP is improved from 43.5% to 45.2%. The YOLOv5 [75] as seen in the figure 3.5 outperforms other models except that Faster R-CNN [111] has 0.1 mAP higher value at 4000 steps for MS-COCO [81] dataset.

With all the enhanced results for object detection and localization, by adopting these newer versions of YOLO [107] family of models, there is a potential scope of improvement for obtaining good performance with regards to speed and accuracy and achieving real time performance simultaneously. Therefore, these novel methods YOLOv5 [75], YOLOR [130] and PP-YOLO [84] were selected for our polyp detection and localization task for thesis work.

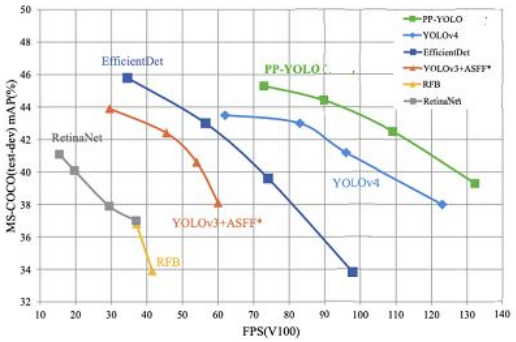

Figure 3.4: Comparison of PP-YOLO on MS COCO dataset with various models. Image is taken from [84]

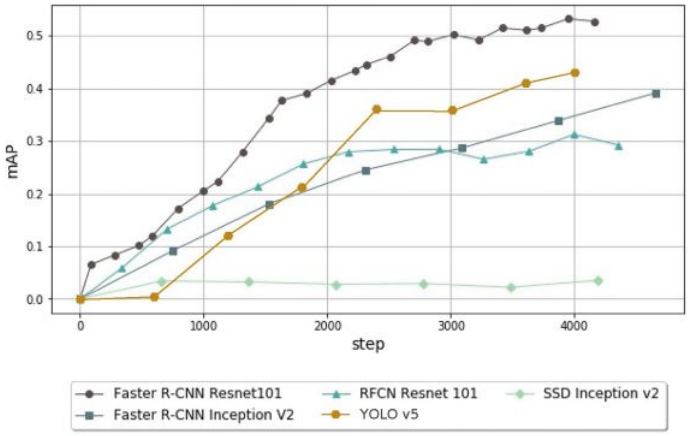

Figure 3.5: Comparison YOLOv5 [75] model with other object detection models. Image is taken from [20]

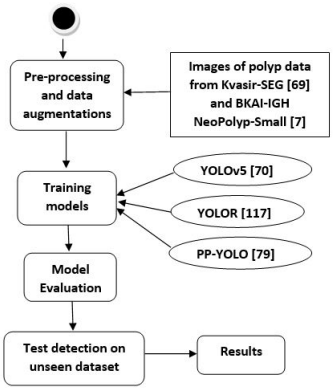

Figure 3.6: Methodology overview of polyp detection and localization task

The figure 3.6 shows the methodology overview of configuration for our polyp dataset for different models YOLOv5, YOLOR and PP-YOLO. The

images from the polyp dataset Kvasir-SEG [74] and BKAI-IGH NeoPolyp-Small [29] would be pre-processed, data augmented and prepared required for the model. Then the training of the models would be carried out followed by evaluation of the model performance and performing polyp detection and localization on unseen test datasets. To perform the experiments for this thesis, the existing models YOLOv5 [75], YOLOR [130] and PP-YOLO [84] models were chosen for configuration for custom datasets from endoscopy examinations. The experiments are performed for the three object detection models using the dataset from the colonoscopic examinations in order to evaluate and compare their performance for different parameters and settings. The neural network architecture YOLOV5 [75], YOLOR [130] and PP-YOLO [84] have reported higher accuracy values with regards to the COCO dataset [81] therefore these existing models were selected for doing configurations in order to run the experiments on the Kvasir-SEG [74] dataset.

The process of creating a custom model involves continuous cycle of collecting and organizing input image data, labelling the object which is of interest, training of the model, deploying the model for making the predictions and then using the deployed model to again repeat the same process in order to improve the performance. The following sections describe in detail all the steps required for configuration of the existing all the three models using custom dataset which in this case is Kvasir-SEG [74]. In addition, another independent dataset BKAI-IGH NeoPolyp-Small [29] is used to further retrain the model so that the models learn more polyp object related features and are able to make predictions with better performance.

3.3 YOLOv5

The YOLO [107] model which stands for You Look Only Once as discussed in the section is one of the best object detection family of models with state-of-the-art performances. The company Ultranytics released next version in YOLO [107] family that is YOLOv5 [75] in the year 2020 immediately few days after the release of YOLOv4 [31]. There is a controversy on YOLOv5 [75] because currently there is no peer-reviewed research paper published by its author Glenn Jocher [88]. The implementation of YOLOv5 [75] is a PyTorch extension of YOLOv3 [110] and is easy to use.

Operational viewpoint	YOLOv4	YOLOv5
Installation	Need to build 'darknet.exe' app from the darknet repository	Only run the requirements.txt file to start running the model
Directory Structure	Requires two paths for directories one for images and other for annotations(txt or XML format)	Uses .yml file format
Storage Size	Weights are stored in '.weights' format	Weights are stored in '.pt' format(PyTorch format)

Table 3.1: Comparison between YOLOv4 and YOLOv5 models from operational viewpoint [88]

There are three main blocks in the architecture in the YOLOv5 [75] [88] family of models namely Backbone, Neck and Head.

- **YOLOv5 backbone [88]**: CSPDarknet [129] is used as the backbone to perform feature extraction from images which have cross-stage partial networks.
- **YOLOv5 neck [88]**: PANet [141] is used to generate a feature pyramids network for aggregating on the features and it passes to Head for prediction.
- **YOLOv5 head [88]**: It consists of the layers which generate predictions from the anchor boxes.

For training, the activation functions used by YOLOv5 [75] are leaky ReLU and sigmoid activation and optimization options used are SGD and ADAM. YOLOv5 [75] uses Binary cross-entropy with logits loss [88]. There are multiple options of pre-trained models for YOLOv5 [75] as seen in the Figure. The different models vary in the size and inference time as shown in the figure 3.7. For example, the model YOLOv5s [75] is lightweight model which is 14MB but not that accurate when compared to YOLOv5x [75] model which is most accurate but has bigger size of 168MB.

3.3.1 Configuring YOLOv5 for our polyp segmentation datasets

In our case, we would first train the model using Kvasir-SEG [74] dataset consisting of 1000 images with the corresponding bounding box details

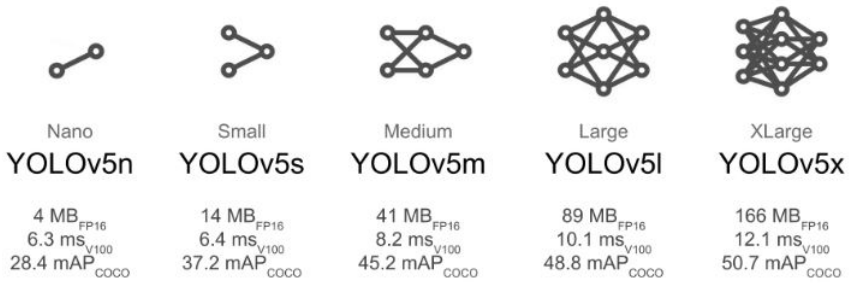

Figure 3.7: Different network models in YOLOv5

in json file indicating the polyp location in the image. The following steps explain in detail to train the YOLOv5 [75] model using our polyp segmentation dataset. The reference and guide for the configurations done for polyp dataset for YOLOv5 model are taken from the github link https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data [**?**]

- The prerequisite pytorch library is installed for YOLOv5 model to run.
- Next, the yolov5 repository [75] in the github link https://github.com/ultralytics/yolov5 is cloned into the working notebook and requirements.txt file is installed for ensuring that the required libraries are installed successfully.

1. Prepare dataset

The YOLOv5 model comes with the sample setup for COCO dataset [81]. The following configurations are done by reusing the config files already existing and making the necessary modifications required for setting up the environment for Kvasir-SEG [74] dataset in YOLOv5 acceptable format.

Create dataset configuration file: A file named *polyp.yaml* is created which is the dataset configuration file that would define the following parameters:

- Dataset directory
  path: path for dataset root directory
  train: relative paths to train image directories or or *.txt files with image paths
  val: relative paths to val image directories or or *.txt files with image paths

test: relative paths to test image directories or or *.txt files with image paths

- Number of classes *nc* - In our case nc=1 as the class object is only one which is *polyp*.
- A list of classes *names*

```
path: /content/datasets/Kvasir-SEG
train: /content/datasets/Kvasir-SEG/images/train
val: /content/datasets/Kvasir-SEG/images/val
nc: 1
names: ['polyp']
```

Figure 3.8: Snapshot of contents of polyp.yaml file

Create labels: The labels for the images need to be reproduced in the YOLO recognized format. For each image, one .txt file is required which denotes the object label for the image. In case of no objects of interest in the image, then no .txt file is required. The .txt file specifications are as follows:

- One row for each object. In case of multiple objects in single image then multiple rows corresponding to the objects labels found in the image.
- Each row has 5 values in the following sequence that is class xcenter ycenter width height format.
- The coordinates for the bounding box should be in the normalized xywh format (value should be from 0 - 1). If the boxes are in pixels then normalized value for xcenter and width would be by dividing xcenter by image image width and dividing width by image width. Similarly, the normalized value for ycenter and height would be by dividing ycenter by image height and dividing height by image height.
- Class numbers are zero-indexed and the numbering starts from 0.

Organize the directories: The train and val images along with their labels for our Kvasir-SEG dataset need to be organized as required for the model to run the training. The directory folder for Kvasir-SEG is created

inside the /datasets directory which is placed next to the /yolov5 directory as shown in the figure 3.9. The labels are automatically located by YOLOV5 by replacing the last instance in each image path of /images/ with /labels/. An example is shown as below:

```
../datasets/Kvasir-SEG/images/im0.jpg  # image
../datasets/Kvasir-SEG/labels/im0.txt  # label
```

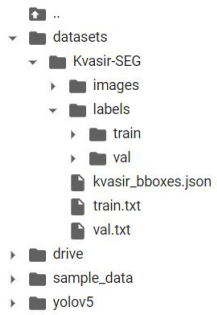

Figure 3.9: Folder structure for dataset setup for YOLOv5 model

Training the the model

The following command is executed to start the training from the defined pre-trained model or not specifiying any parameter which would mean randomly initialized weights in the –weights parameter. However, the specification of pretrained weight is recommended due to the fact that the training need not start from the scratch. The pre-trained weights are automatically downloaded from the latest YOLOv5 release. The image size can be specified with the –img parameter, batch size can be defined using –batch parameter. Similary number of epochs can be given in –epochs and the –data parameter should be provided with the path to the dataset.yaml file which in our case is polyp.yaml file. The training results are saved to the folder runs/train/.

```
python train.py --img 640 --batch 16 --epochs 300
--data polyp.yaml --weights yolov5s.pt
```

3.4 YOLOR Model

YOLOR [130] Object Detection is particularly for object detection when compared to other tasks such as object identification or analysis. Humans learn and understand the surroundings around them through vision, hearing, etc which is explicit knowledge and also from the past experiences which is implicit knowledge. Hence, due to this, humans have the ability to process completely new data with the use of vast experience and learning which is stored in the brain. Based on this concept, the researchers in YOLOR paper [130] proposed an approach which utilizes both explicit knowledge defined as learning from the given input data and implicit knowledge which is learned throughout subconsciously.

Architecture of YOLOR The architecture of YOLOR [130] comprises of three functional process:kernel space alignment, prediction refinement, and a convolutional neural network (CNN) with multi-task learning. The results demonstrate that if the neural network is provided with implicit knowledge in addition to the network already trained with explicit knowledge, it benefits in terms of improving the performance of different tasks.

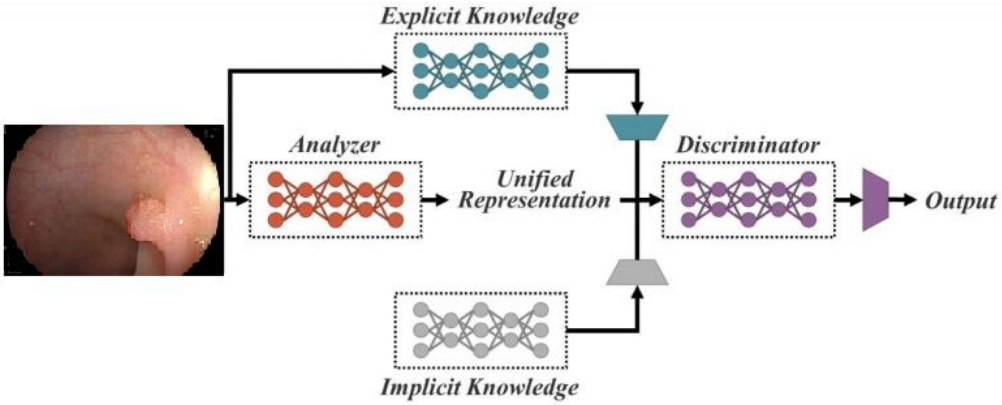

Figure 3.10: YOLOR Model with implicit and explicit knowledge-based multi-task learning. Image is taken and modified from [130]

3.4.1 Configuring YOLOR for our polyp segmentation dataset

The reference for configuration of YOLOR model is taken from the online youtube tutorial [137] and from the github link [138]. The model is configured with our custom dataset Kvasir-SEG [74] by modifying the necessary model parameters suitable for our application scenario. The configuration of YOLOR model requires the following steps:

- Prerequisites: The Pytorch and Cython needs to be installed before starting the model training. Pytorch library is required as YOLOR is implemented using Pytorch library and should be installed according to the GPU and CPU compatibility.

- Clone the repository: The following github repository should be cloned [138].

  `https://github.com/WongKinYiu/yolor`

- Create polyp.names and polyp.yaml files for our dataset: The polyp.names file would list the names of the object classes. In our case it is 'polyp'. The polyp.yaml will contain the train and test image paths, number of classes and class names as shown in the figure 3.11.

```
# train and val datasets (image directory or *.txt file with image paths)
train: /content/yolor/dataset/train.txt
test: /content/yolor/dataset/test.txt

# number of classes
nc: 1

# class names
names: ['polyp']
```

Figure 3.11: polyp.yaml file contents for YOLOR.

- Modify the model parameters in the connfiguration file: The YOLOR configuration file yolor_p6.cfg is based on the COCO dataset [81]. Therefore this file should be modified according to our dataset requirements.The changes for the following parameters are required in the configuration file:

– *classes*: The file contains the 'classes' parameter as set to 80 for COCO dataset, therefore it needs to be changed to 1 as we have one object class 'polyp' everywhere in the file.

– *filters*: The file contains the 'filters' value as 255. It needs to be changed to the value of 18 which are just above the classes parameter according to the formula as below. $(number of classes + 5) \times 3$ As we have number of classes 1, therefore the filters value comes as 18.

- Train the model: The following command is executed to train the algorithm. The best weight file and last weight file for last epoch run is generated after the training is completed.

```
python train.py --batch-size 8
--img 1280 1280 --data custom.yaml
--cfg cfg/yolor_p6_custom.cfg
--weights '' --device 0 --name yolor_p6
--hyp hyp.scratch.1280.yaml --epochs 300
```

3.5 PP-YOLO Model

PP-YOLO [84] stands for PaddlePaddle – You only look once. This framework simplifies the process of object detection in construction, training, optimization and deployment in a faster and efficient way by providing end-to-end methods. It provides various pre-trained models for object detection, instance segmentation, face detection, etc. It enables the developers to construct different pipelines fast with modular designs. It also supports distributed training.

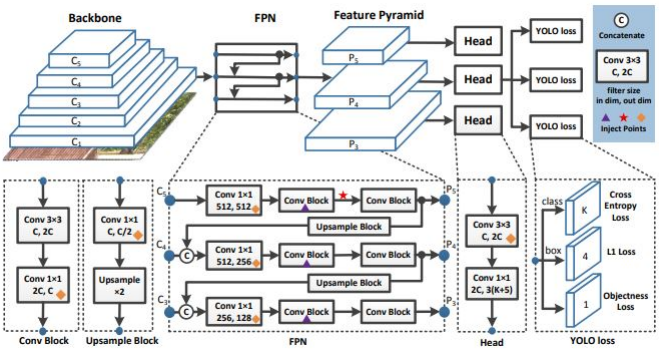

Figure 3.12: The network architecture of YOLOv3 and inject points for PP-YOLO. Image is taken from [84]

Architecture of PP-YOLO The architecture of PP-YOLO [84] is mainly based on YOLO3 [110]. The architecture is divided into three categories as follows:

- **Backbone [84]:** It contains the convolution neural network which is a pre-trained classification model for generating features.
- **Detection Neck [84]:** Feature Pyramid Network(FPN) is created with a pyramid of features by combining and mixing the ConvNet representations.
- **Detection Head [84]:** It does the prediction and bounding box on the image object.

3.5.1 Configuring PP-YOLO for polyp segmentation datasets

The reference for configuration steps required of PP-YOLO [84] model is taken from the tutorial [8]. The model is configured with our datasets Kvasir-SEG [74] and BKAI-IGH NeoPolyp-Small [29]. The configuration of PP-YOLO model involves the following steps:

- Install PaddlePaddle: The commands to install PaddlePaddle and to verify that it is installed successfully are as follows

```
!pip install paddlepaddle-gpu
import paddle
paddle.utils.run_check()
```

- Install Paddle Detection: For installing paddle detection, the following github repository needs to be cloned.

```
https://github.com/PaddlePaddle/
PaddleDetection.git
```

- Install other dependencies, paddledet and pycocotools. After the installation, check whether the tests pass through the below command.

```
python ppdet/modeling/tests/
test_architectures.py
```

1. Prepare dataset

For pointing the objects of interest in an image, it is necessary to mark the position and category for each object. Below are the three different ways in which the object position is represented by a rectangular box also known as bounding box.

Expression	**Explanation**
x1,y1,x2,y2	(x1,y1) is top left coordinate point,(x2,y2) is bottom right coordinate point
x1,y1,w,h	(x1,y1) is the top left coordinate point, w is width and h is height of object
xc,yc,w,h	(xc,yc) is center coordinate of object, w is width and h is height of object

Table 3.2: Different object position representations for bounding box. Table is taken from [48]

The paddle detection supports COCO [81], PASCAL VOC [54] and wider-face [142] datasets by default. In our case, we would convert our datasets Kvasir-SEG [74] and BKAI-IGH NeoSmall-Polyp [29] to VOC format. The Pascal VOC dataset uses the notation of [x1,y1,x2,y2] for representing the bounding boxes for an object. The structure for organizing the files for VOC dataset is as follows:

- label_list.txt this is the list of classes name.

```
>>cat label_list.txt
polyp
```

- trainval.txt is the file consisting of list of trainset which contains the rows for each image having two values, first one containing the path to the image and the second containing the path to the annotations for the corresponding image.

```
>>cat trainval.txt
./JPEGImages/img0.jpg ./Annotations/img0.xml
./JPEGImages/img1.jpg ./Annotations/img1.xml
```

- test.txt is file list of testset

```
>>cat test.txt
./JPEGImages/img5.jpg ./Annotations/img5.xml
./JPEGImages/img6.jpg ./Annotations/img6.xml
```

For dataset to be in VOC format, there is an annotation file in XML format containing the details for object bounding boxes for each image file having the same file name as image file name. The XML file has the below fields

- *filename*: Specifying the image name.

- *size*: Specifying the image size including width, height and depth of the image.

```
<size>
    <width>500</width>
    <height>375</height>
    <depth>3</depth>
</size>
```

Convert user data into VOC format. Once the dataset for Kvasir-SEG [74] and BKAI-IGH NeoSmall-Polyp [29] is converted into the voc format which is recognized by PP-YOLO [84] then the directory structure looks as in the figure 3.13.

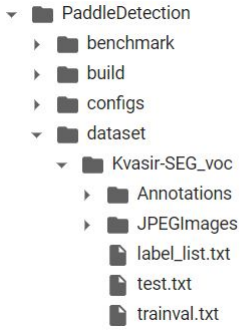

Figure 3.13: Directory structure for dataset setup for PP-YOLO model

YOLO model parameter configuration The parameter configuration for the PP-YOLO consists of five sub profiles as below. The changes are required in the data profile, data reads and runtime file for the required settings for data augmentations and hyperparameter configurations.

- Data profile coco_detection.yml
- Optimizer configuration file optimizer_1x.yml
- Data reads configuration files ppyolo_reader.yml
- Model profile ppyolo_r50vd_dcn.yml
- Runtime file runtime.yml

To start the training of the model the below command is executed.

```
python tools/train.py
-c configs/ppyolo/ppyolo_r50vd_dcn_Kvasir-SEG_voc.yml
```

3.6 Experimental setup: dataset split

One of the important task in object detection is to identify a smart way to split the given dataset into the train and validation sets. While distributing the object detection dataset into train and validation, it is important to take into consideration the distribution of the objects identified in the image rather than the images themselves.

In our case of Kvasir-SEG [74] dataset and BKAI-IGP NeoSmallPolyp [29] where all the images contain only one type of object label that is

a polyp. Normally, the standard split ratio of the images in 80/20 randomly is recommended where 80 is for train and validation and remaining 20 is for independent test dataset. In this case, the already defined train-val split split in Medico automatic polyp segmentation task challenge is considered. The train.txt and val.txt files containing the list of image name are available in the github link https://github.com/DebeshJha/2020-MediaEval-Medico-polyp-segmentation for fair comparison of different models [70].The train.txt contains 880 images and val.txt contains 120 images.

The basis for considering this split is that the earlier results from the state-of-the-art models related to object detection tasks with the same Kvasir-SEG [74] dataset had been obtained on the same train-val split. Therefore, it would easier for comparison of the implemented models in this thesis with the earlier benchmark models for performance metrics evaluation.

3.7 Data augmentation

In order to train the neural networks and obtain the best performing model, it requires larger amount of data. However, when it comes to the field of medicine where medical images are required, it is not always feasible to collect huge amount of dataset. Therefore, if the neural network needs to be developed effectively to classify medical images with limited dataset, then one of the ways is to perform augmentation on the available dataset and utilize the transformed images for the training the model. Data augmentation often has an improvement in the performance of deep learning neural network with amount of data available due to additional labelled training data and different orientation of the images for learning more image features. The model implementation for YOLOv5 [75], YOLOR [130] and PP-YOLO [84] come with default parameter value setting for various data augmentation techniques. These values are obtained and set based on several iterations of training the models giving best evaluation results on larger datasets such as COCO dataset [81]. Below presents the data augmentation techniques that would be utilized during training the model.

- Mosaic [53]: This technique works by combining four input source images into a single image. The strategy for combining the images

is as follows:

  - Simulates four random crops while ensuring the relative scale of the objects in comparison to the image.
  - Combines different classes for the objects that are not found together in a single image in the training dataset For example, if there are images of cats and images of dogs, but no images of cats and dogs together in the training data, then mosaic augmentation would simulate it.
  - It simulates the variation of the number of objects found in the images. For instance, if the images contains only one bounding box denoting one object then the mosaic augmentation would output between zero and four bounding boxes.

- **Flipping** [125]: An image flip means mirroring the image either on the horizontal or vertical axis.
- **Rotation** [125]: An image rotation would rotate the image clockwise from 0 to 360 degrees.
- **Brightness** [125]: An image can be augmented through randomly darkening the images, brightening the images or both.This would help the model to generalize across images with different levels of lighting.
- **Cropping** [125]: This would pad the image so that black pixels are added around the image and then crop the image randomly to the original size of the image.
- **Zooming** [125]: This augmentation randomly zooms in or zooms out the image by either adding new pixel values around the image or interpolating the pixel values.

3.8 Hyperparameters settings

The training of the model for the neural network requires lot of computational resources and is time consuming. Therefore the optimal selection of parameters required for efficient training process is essential. Following are the parameters considered to train the detection model using YOLOv5 [75], YOLOR [130] and PP-YOLO [84].

3.8.1 Batch Size

The choice of batch size depends mainly on how large is the size of the neural network and what are the memory usage availability for execution. The batch size is a hyper-parameter that defines the number of training samples to work with before the internal parameters of the model are updated. After completion of each batch processing, the predictions are compared with the expected output and error calculation is done. The learning algorithm of the training model is classified as follows according to the way of dividing the training dataset for processing [35]:

- **Batch Gradient Descent** Batch Size Training dataset size
- **Stochastic Gradient Descent** Batch Size = 1
- **Mini-Batch Gradient Descent**. 1 < Batch Size < Training dataset size

The mini-batch gradient descent is commonly used and recommended approach for batch size selection in deep learning. The benefit of mini-batch gradient descent is that the memory required for processing is less and network training is faster [35]. The dataset in our case is fairly large and therefore practically impossible to go for batch gradient descent approach, therefore mini-batch gradient is suitable for training the model in our case. The research papers [27], [90] recommend the batch size of 32 as appropriate. Therefore, the batch size of 32 would be used for training the models.

3.8.2 Number of epochs

This hyperparameter defines the number of iterations for which the training algorithm would process the complete training dataset. The training for one epoch means that every input sample in training dataset was able to update the internal parameters for the model [33]. After experimenting for different epoch values and observing the loss, accuracy and performance results, the number of epochs chosen for our experiments is 300. This is to avoid over-fitting as well as for increasing the generalization in training the neural network.

3.8.3 Weights initialization

The initialization of weights is a key parameter when designing and implementing the neural networks. The neural network model is designed to fit with an optimization algorithm known as stochastic gradient descent. This algorithm modifies the weights in the network for minimizing the value of the loss function for arriving at a set of weights that can make good predictions. Neural network models are fit using an optimization algorithm called stochastic gradient descent that incrementally changes the network weights to minimize a loss function, hopefully resulting in a set of weights for the mode that is capable of making useful predictions. Initial starting points for the possible weights values need to be given for the optimization algorithm to begin with.The weights initialization for a neural network model is to set the initial weights for the starting point for training the model [92].

If the weights are initialized improperly then it can have negative impact on the training process due to either vanishing or exploding gradient issue. In case of vanishing gradient problem, the update in weights is minor and thereby having slower convergence or in worst cases due to slowness in optimization algorithm it may completely stop the convergence. The exploding gradient problem happens during forward or back-propagation when the weight initialization is too large [92]. The weights initialization for the training models would be done using the pre-trained weights trained on ImageNet instead of initializing the weights to a random or from scratch.

3.9 Performance evaluation metrics

The performance evaluation metrics provides the measure for validating the performance of the model on the task of object detection and also enables to do comparison with the other state-of-the-art detection models. The following useful metrics would be used to evaluate the performance of the object detection/localization model.

Unlike classification task where it only evaluates the probability of the object present in the image, the object detection task needs to localize the object along with outputting the bounding box for each object and its corres-

ponding confidence score indicating the certainty of the bounding box over the object being detected. Hence, the metric Intersection over Union (IoU) is used for determining the number of objects detected correctly or incorrectly.

Intersection over Union (IoU) Intersection over Union is also known as the Jaccard Index is an evaluation metrics that calculates the overlap between the ground truth annotations(gt) (target objects that are annotated with bounding boxes in test dataset) and the predicted bounding box(pd). The shape of the ground-truth and prediction can be any such as rectangular, box, circle or an irregular shape. The IoU score value is from 0 to 1, if the two boxes are closer to each other then it has a higher IoU score value. According to its mathematical definition, it is the area of intersection divided by area of union of the ground-truth bounding box and predicted bounding box.

$IoU = area(groundtruth \cap prediction)/area(groundtruth \cup prediction)$

For determining the performance of the object detection model, the key thing is whether the detection is correct or not. The threshold value is set for each detection by calculating the IoU score for every detection. If the IoU values are more than the threshold value, then the detection are considered as positive predictions and for those which are below the threshold value are considered as false predictions.

The predictions are mainly classified into true positive (TP), false positive (FP), false negative (FN) and true negative (TN) with respect to object detection and localization task. TP is correct detection performed by model with IoU above the threshold value. FP is an incorrect detection predicted by the object detection model where the predicted box detects IoU value below the threshold value against ground truth box or in another case where object is not present but the model detects an object. Similarly, FN is the ground-truth is missed or not detected by the object detection model. Lastly, TN is a situation where empty boxes that are not explicitly annotated are correctly detected as non-object.The model would identify multiple empty boxes which does not help in adding value to the algorithm. Therefore, this metric is not used in object detection tasks.

If the threshold for IoU is t then detection for True Positive(TP) is where IoU(gt,pd) $>=$ t and False Positive(FP) is where IoU(gt,pd) $<$ t. False

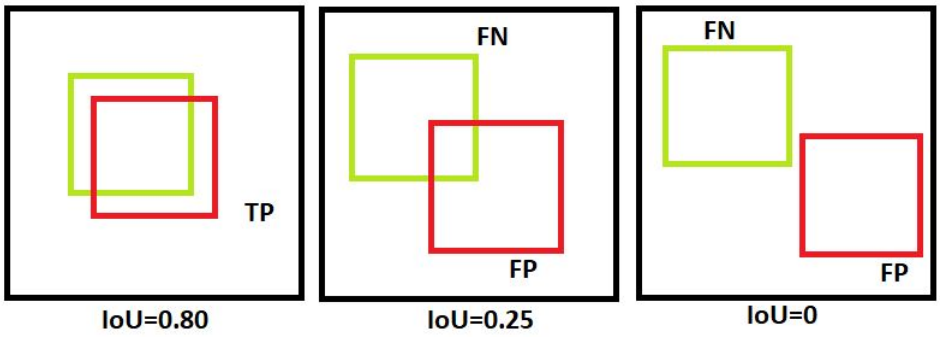

Figure 3.14: IoU Thresholding

Negative(FN) is the case where the ground-truth is completely missed by predicted box. The figure illustrates the scenario of identifying TP,FP and FN where IoU threshold t = 0.5.

Precision:It is the measure of correctness in predictions to identify only relevant objects. It is calculated as the ratio of all True Positives over the entire detections done by the model.

Recall:To calculate the true predictions from all correctly predicted data. It is calculated as the ratio of all True Positives over all the ground truths.

Precision-Recall (PR) curve:It is the plot of precision versus recall for different values of confidence scores. The precision and recall values are higher for a model with good performance even when confidence scores are changed.

Average Precision (AP): Based on the precision-recall curve, Average Precision AP summarises the weighted mean of precisions for each threshold with the increase in recall. AP is calculated for each object class.

Mean Average Precision (mAP): It is an extension of Average precision. In AP, calculation is done only for individual object classes but mAP provides the precision for the entire model. For obtaining the percentage of correct predictions in the model, mAP is used.

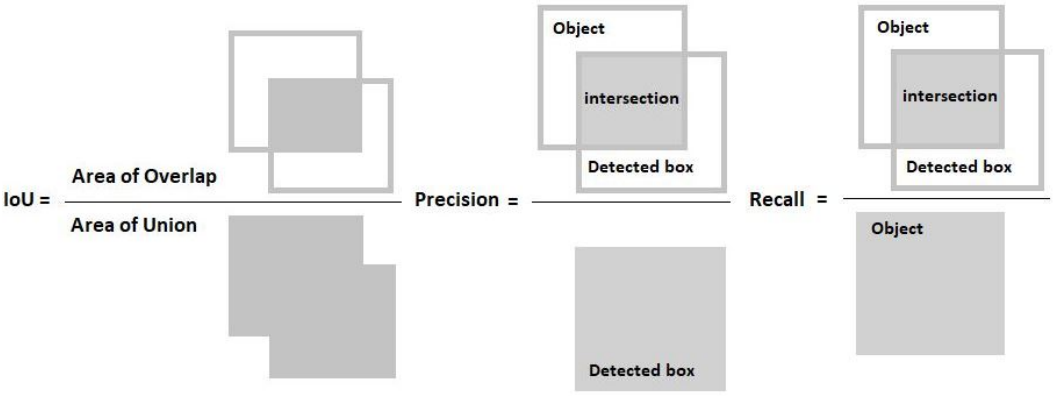

Figure 3.15: IoU, Precision and Recall

3.10 Summary

This chapter discussed the datasets to be used in the thesis work and the reasons behind choosing the models YOLOv5, YOLOR and PP-YOLO models. Further, it discussed in detail the configuration steps required for implementation of our application scenario of detection and localization of polyps in images. Also, the various hyperparameter settings and performance evaluation metrics were described. In the next chapter, the results obtained from various experiments would be presented.

Chapter 4

Results

This chapter presents the results obtained from various experiments conducted in this thesis work. In details, it lists the different results obtained for each model implemented YOLOv5 [75], YOLOR [130] and PP-YOLO [84] using custom datasets Kvasir-SEG [74] and BKAI-IGH NeoPolyp-Small [29] and with all the variations done for experiments. It includes running the model on base dataset, with some data augmentations and training the model to include the input dataset without any polyp object to serve the purpose of including negative training cases. Also, the observations from the various results obtained are discussed. The chapter ends with a summary of the results and general observations from the experiments.

The python execution codes and files for all the models are available at the github link https://github.com/AMITAKASHIKAR/Deep-Learning-models-for-polyp-detection-and-localizationaset and also in the appendix.

4.1 Experiments with YOLOv5 model

Multiple experiments were performed for YOLOv5 [75] model to validate its performance for different parameters. Firstly the YOLOv5 [75] model consists of several types of network model as described in the section :YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. According to the , YOLOv5s is the fastest and reliable models available which have been tested on MS COCO dataset [81]. The experiments were performed for 300 epochs for each of the models. The model experiment with YOLOv5x could

YOLOv5 network model	**mAP_0.5**	**Precision**	**Recall**
YOLOv5n (nano)	0.8913	0.9075	0.8307
YOLOv5s (small)	**0.8946**	**0.9316**	0.8382
YOLOv5m (medium)	**0.8946**	0.9051	0.8076
YOLOv5l (large)	0.8937	0.9091	**0.8461**

Table 4.1: Training results for different YOLOV5 [75] network models.

YOLOv5s image size	**Execution time in seconds**	**mAP_0.5**	**Precision**	**Recall**
640	5596s	0.8946	0.9316	0.8382
1280	21552s	0.8498	0.8491	0.7846

Table 4.2: Training results metrics for YOLOv5s for different image sizes

not be carried out as the time and memory usage required for the experiment for sufficiently large and the intention was to showcase the trade-off between model performance and time and memory required. After performing the experiments for Kvasir-SEG [74] for polyp detection on the the different YOLOv5 models, the results obtained are shown as in the table 4.1.

As seen in the table 4.1, the values for precision 0.9316 was highest for yolov5s. The recall value was highest for yolov5l model which was 0.8461. However, an important decisive factor to note here is that yolov5l required significantly more execution time and memory when compared to yolov5s. Therefore, for running further experiments on YOLOv5 [75] for our dataset, the model yolov5s is selected due to its faster results and fairly good performance.

The table 4.2 shows the results when the YOLOv5s model is trained for image resolution size 640 and image resolution size 1,280. The results clearly show that when the image resolution is higher than the execution time for training increases significantly. The time taken for image size 640 is 5,596 seconds whereas for image size 1,280 it is 21,552 seconds. The precision, recall and mAP_0.5 value for image size 640 is 0.931, 0.838 and 0.894 whereas

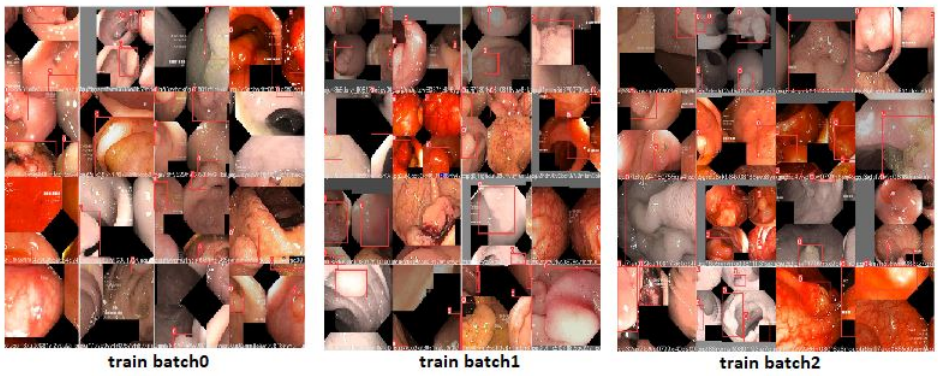

Figure 4.1: Images of train batch for YOLOV5 training using base dataset

for image size 1280 it is 0.849, 0.785 and 0.849. With the increase in the image resolution, the performance of the model is reduced.

The experiment was performed using base dataset Kvasir-SEG [74] having 1,000 polyp images containing bounding box values. All of the images contain either one or more polyp object. By default, YOLOv5s [75] model does mosaic data augmentation along with default parameters set for various augmentation attributes in hyperparameter config file.

4.1.1 Training the Model

The figure 4.1 shows the batch of images with several internal data augmentations used by default in YOLOv5 [75] model for training the model. The YOLOV5 [75] model is comprised of 270 layers, 7022326 parameters, 7022326 gradients and 15.8 GFLOPs. However, the model was trained using 213 layers, 7012822 parameters, 0 gradients and 15.8 GFLOPs as number of classes (nc) nc=80 was ovverrided by nc=1 since the number of classes in this case is 1 which is *polyp*.

The figure 4.2 shows the model summary of YOLOv5 model during the model training. The optimizer used for training the model using yolov5s.pt (small network) model is Stochaistic Gradient Descent (SGD). The time taken for training the model YOLOv5 for 880 images in train dataset and 120 images in validation dataset for 300 epochs with batch size 32 was 1.555

```
Overriding model.yaml nc=80 with nc=1

                 from  n    params  module                                  arguments
  0                -1  1      3520  models.common.Conv                      [3, 32, 6, 2, 2]
  1                -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
  2                -1  1     18816  models.common.C3                        [64, 64, 1]
  3                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  4                -1  2    115712  models.common.C3                        [128, 128, 2]
  5                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  6                -1  3    625152  models.common.C3                        [256, 256, 3]
  7                -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
  8                -1  1   1182720  models.common.C3                        [512, 512, 1]
  9                -1  1    656896  models.common.SPPF                      [512, 512, 5]
 10                -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12          [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  1    361984  models.common.C3                        [512, 256, 1, False]
 14                -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16          [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  1     90880  models.common.C3                        [256, 128, 1, False]
 18                -1  1    147712  models.common.Conv                      [128, 128, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  1    296448  models.common.C3                        [256, 256, 1, False]
 21                -1  1    590336  models.common.Conv                      [256, 256, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  1   1182720  models.common.C3                        [512, 512, 1, False]
 24      [17, 20, 23]  1     16182  models.yolo.Detect                      [1, [[10, 13, 16, 30, 33, 23],
Model summary: 270 layers, 7022326 parameters, 7022326 gradients, 15.8 GFLOPs
```

Figure 4.2: Model summary of YOLOv5 [75] during model training

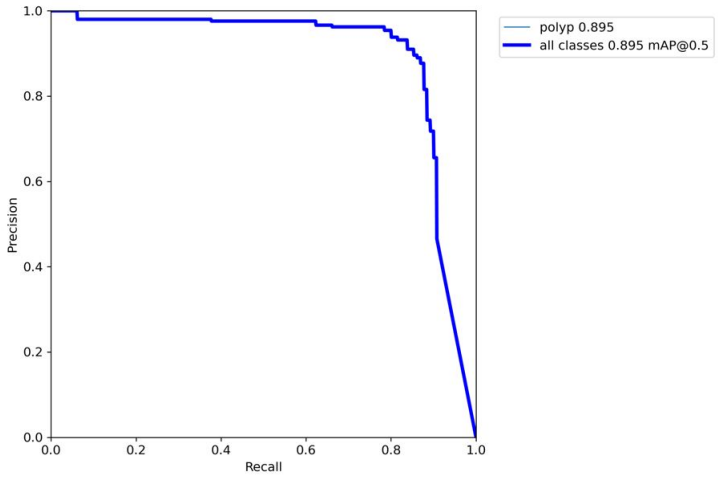

Figure 4.3: PR Curve for YOLOv5 model training with Kvasir-SEG base dataset [74] during model training

hrs. The precision value was 0.9316 and recall value was 0.8382.

The figure 4.3 shows the PR curve for the YOLOv5 model training for the Kvasir-SEG [74] base dataset. The graph shows a decent trade-off between the precision and recall values indicating a good amount of area under the PR-curve having the mAP@0.5 value of 0.895.

The figure 4.4 displays the plots for box loss, object loss, classification

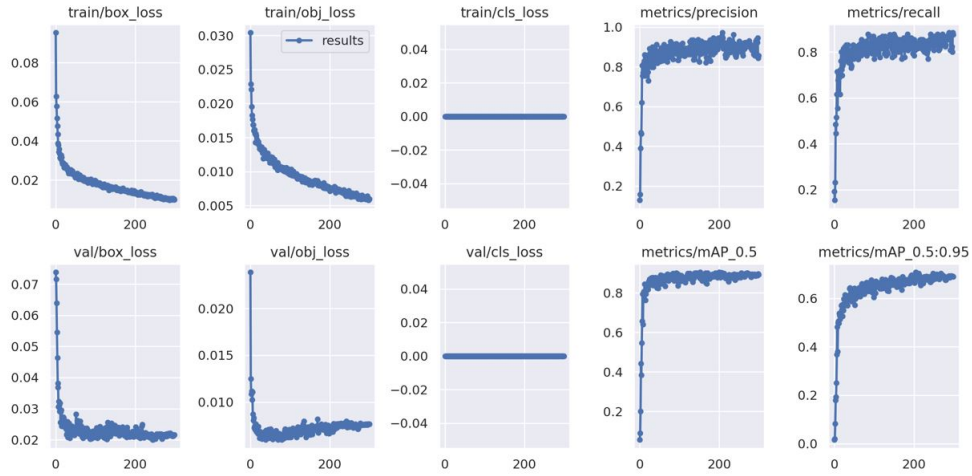

Figure 4.4: Plots of box loss, object loss, precision, recall and mAP value for YOLOv5 training over number of epochs

loss for train and validation sets. The box loss during the prediction of the bounding box is calculated as the error between the predicted bounding box and the ground truth bounding box [127]. This actually means that the less the value of the box loss, the more accurate is the algorithm in predicting the bounding box having maximum overlap with the ground truth bounding box. The object loss is denoted as how confident is the algorithm to locate the presence of an object [25]. The model rapidly showed an improved values for precision, recall and mAP after about 50 epochs. Also, the box loss and object loss showed fast decline after around 25 epochs.

4.1.2 Evaluating the model

The validation of the model is done on the 120 images from the validation dataset for three iou threshold values IoU=0.25, IoU=0.50 and IoU=0.75 using the best weights obtained from the model training. The table 5.4 illustrates the performance results obtained from the evaluation of the model. The results suggest that when the IoU threshold value is increased, there is a reduction in the precision value from having 0.92 precision at IoU threshold value of 0.25 to 0.885 precision value at IoU threshold value of 0.75 while the recall value of 0.885 remains nearly constant at all threshold values. Similar behaviour is observed in case of mAP values where there is reduction to 0.888 value for IoU threshold at 0.75 from 0.906 for IoU=0.25 and 0.903 for

IoU threshold value	Precision	Recall	mAP_0.5	Speed in FPS
0.25	0.92	0.885	0.906	102.04
0.50	0.905	0.885	0.903	111.11
0.75	0.885	0.884	0.888	106.38

Table 4.3: Results obtained from validation of YOLOv5 model at different IoU threshold values

IoU=0.5. In case of inference speed, similar characteristics are observed where it is highest for IoU=0.25 and IoU=0.5 having 9.8ms and 0.97ms inference speed to decreased speed of 7.5ms for IoU=0.75.

The figure 4.5 shows the validation batch labels and their corresponding predictions in red colour boxes done on evaluation of the YOLOv5 model with base dataset training.

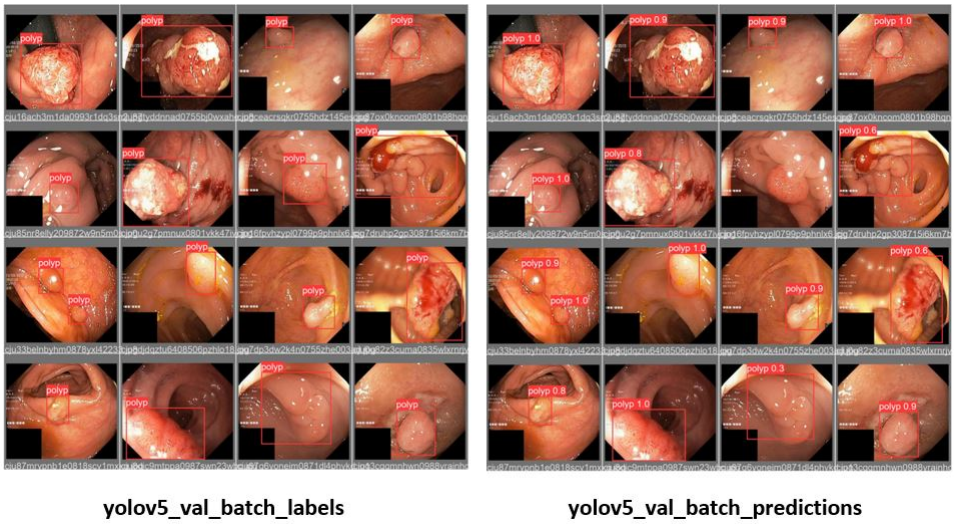

Figure 4.5: YOLOv5 validation batch labels and corresponding predictions

4.2 Experiments with YOLOR model

4.2.1 Training the Model

Training of the YOLOR [130] model using 1000 images from the Kvasir-SEG dataset [74] with their bounding boxes annotations was completed in 3.827 hrs. The YOLOR Model Summary is as follows 665 layers, 36838416 parameters, 36838416 gradients. The following metrics was achieved after

IoU threshold value	Precision	Recall	mAP_0.5	Speed in FPS
0.25	0.796	0.877	0.868	69.44
0.50	0.72	0.892	0.879	68.96
0.75	0.56	0.892	0.682	69.44

Table 4.4: Results obtained from validation of YOLOR model at different IoU threshold values

training was completed with image size of 640x640 and batch size of 16. Precision 0.715, Recall 0.885, mAP@.5 0.895 with speed of 10.5ms for inference, 1.4ms for NMS and total speed of 11.9ms for each image of size 640x640.

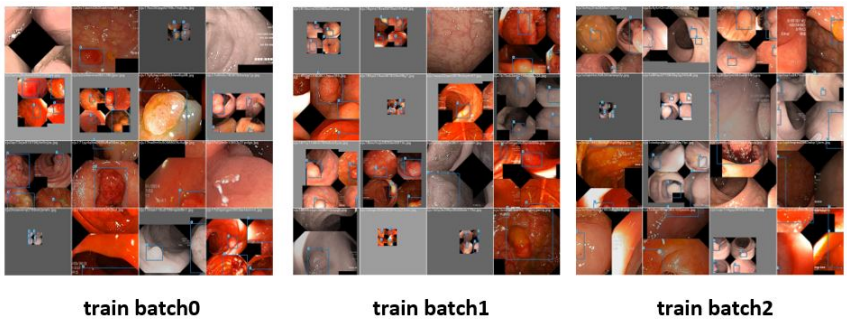

Figure 4.6: Images of train batch for YOLOR training using Kvasir-SEG base dataset

4.2.2 Evaluating the Model

The results table 4.4 for YOLOR model validation for different IoU threshold values shows that there is an increase in the precision value when the IoU threshold value is lower, when IoU=0.25 the precision value is 0.796 whereas when IoU=0.75, the precision value is 0.56. Also, we observe that there is an effect on the mAP value, for higher IoU value, the mAP value is lowest. In this case for IoU=0.75 value is 0.682 which is the least.

4.3 Experiments with PP-YOLO model

The model training using PP-YOLO took total time of nearly 7 hrs for completion. Training with the base dataset for Kvasir-SEG [74], the following

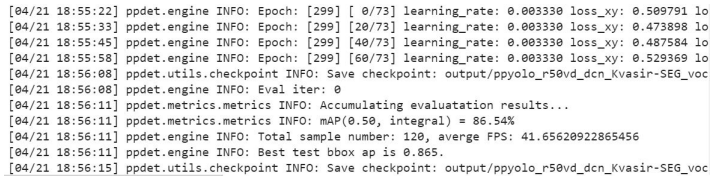

Figure 4.7: Snapshot of PP-YOLO training for Kvasir-SEG base dataset

results were obtained after 300 epochs in the output during evaluation.

```
mAP(0.50, integral) = 86.54%
Total sample number for validation: 120
average FPS: 41.656
Best test bbox ap is 0.865.
```

4.4 Dataset consisting images without polyps

The experiment was again performed for model training with normal images not containing polyps for all the three models. This was done for reducing the number of false positives and the to train the model with more features from the gastrointestinal tract to potentially enhance the model performance.

The training of YOLOv5 for polyp dataset for normal images obtained the results: mAP_0.5 = 0.89526, Precision = 0.95363, Recall = 0.79845

The training of YOLOR for polyp dataset for normal images obtained the results: mAP_0.5 = 0.90174, Precision = 0.78809, Recall = 0.88462

Training PP-YOLO for dataset containing normal images, the following training results were obtained after 300 epochs: mAP(0.50, integral) = 89.53%, Total sample number for validation: 120, average FPS: 41.505, Best test bbox ap is 0.897.

From the experiment of training the model with polyp image samples, we can infer that the model training with more dataset improves the overall performance of the model as we can observe there is some improvement in the mAP and precision value for the models.

Model	IoU threshold value	Precision	Recall	mAP_0.5	Speed in FPS (Frames Per Second)
YOLOv5	0.25	0.79	0.702	0.724	121.96
YOLOv5	0.50	0.781	0.702	0.721	106.38
YOLOv5	0.75	0.781	0.687	0.702	120.48
YOLOR	0.25	0.492	0.755	0.617	76.34
YOLOR	0.50	0.442	0.78	0.628	74.62
YOLOR	0.75	0.347	0.795	0.612	75.18
PP-YOLO	0.50	0.885	0.884	0.888	45.04

Table 4.5: Results obtained after running detection on MediaEval-Medico-polyp-segmentation dataset [70] for models at different IoU threshold values

4.5 Polyp detection on test dataset

The table 4.5 shows the results obtained on running detection for all the three models on MediaEval-Medico-polyp-segmentation test dataset containing 200 images [70]. The ground truth segmented masks are available for the images. However, a small program was written to convert the segmented mask for the images and get the bounding boxes values for label annotations for polyp object. The results indicate that the highest FPS was obtained for YOLOv5 model, followed by YOLOR and YOLOv5. Also, in terms of accuracy, the PP-YOLO model achieved the highest precision, recall and mAP values whereas the YOLOR model achieved the least accuracy among the three models. We discuss the best, average and worst case detection scenarios for the three models.

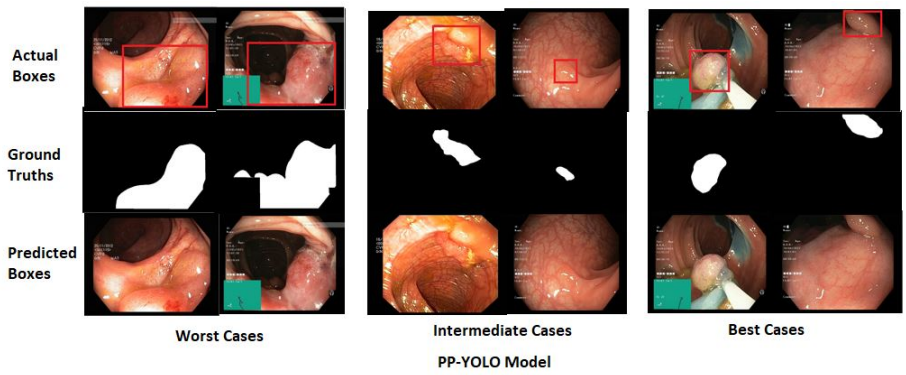

Figure 4.8: Images of worst, average and best scenarios for PP-YOLO

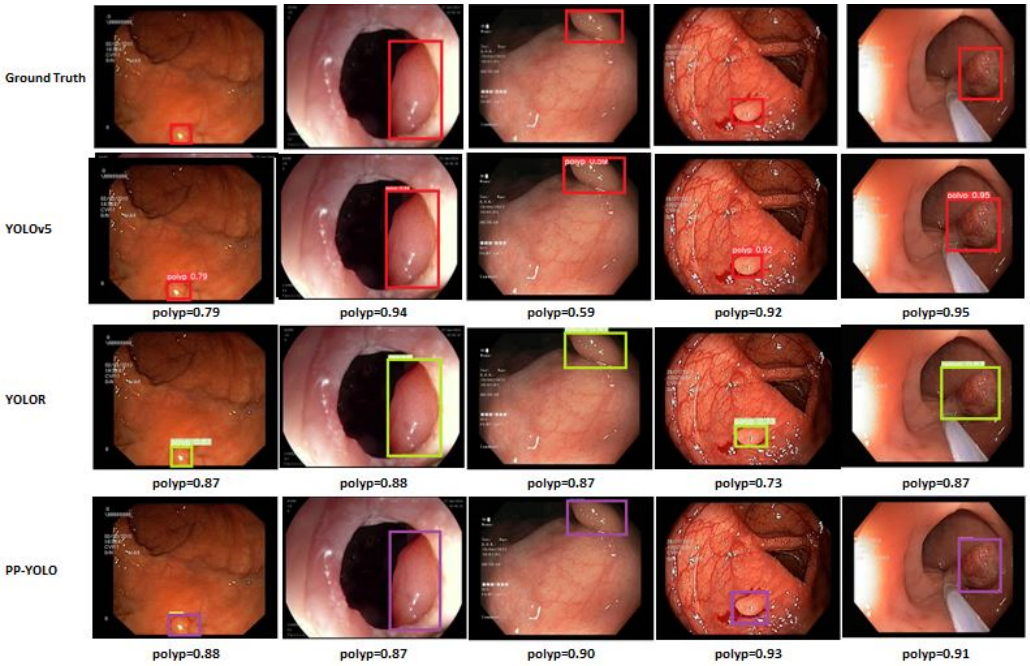

Figure 4.9: Detection Results on test dataset showing bounding boxes predictions

Model	**mAP_0.50**	**Precision**	**Recall**	**Speed in FPS**
YOLOv2	0.7993	0.81	0.78	60
YOLOv3	0.8288	0.94	0.77	40
YOLOv4	0.8372	0.88	0.80	40
YOLOv5	0.702	0.781	0.687	94.33

Table 4.6: Performance comparison for YOLO family of models

4.6 Model performance comparison for YOLO family of models

The table 4.6 shows the comparison results of performance metrics for different versions of YOLO family for our dataset Kvasir-SEG [74]. The results show that the mAP value is best obtained for YOLOv3 model which is 0.8372. However, the precision was highest with 0.94 for model YOLOv3 and lowest for model YOLOv5. The recall value was highest for YOLOv4 with 0.80 and YOLOv5 had the lowest recall value of 0.687, However, in terms of speed YOLOv5 had FPS of 94.33 which was significantly higher than the rest of the models having 60 and 40 FPS. With this comparison, it

indicates that YOLOv5 proves to be the best among all the models in terms of high speed value and with a slightly less accuracy when compared to other models.

4.7 Polyp detection on a video dataset

The models were also tested on a video dataset to visualize how the model behaves when the input is a video of an colonoscopic examination for YOLOv5 model.

```
!python /content/yolov5/detect.py --weights /content/yolov5/best.pt --source /content/yolov5/polypp13.avi

detect: weights=['/content/yolov5/best.pt'], source=/content/yolov5/polypp13.avi, data=data/coco128.yaml,
YOLOv5 🚀 v6.1-177-gd059d1d torch 1.11.0+cu113 CUDA:0 (Tesla P100-PCIE-16GB, 16281MiB)

Fusing layers...
Model summary: 213 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
video 1/1 (1/2238) /content/yolov5/polypp13.avi: 512x640 1 polyp, Done. (0.015s)
video 1/1 (2/2238) /content/yolov5/polypp13.avi: 512x640 1 polyp, Done. (0.010s)
video 1/1 (3/2238) /content/yolov5/polypp13.avi: 512x640 1 polyp, Done. (0.011s)
video 1/1 (4/2238) /content/yolov5/polypp13.avi: 512x640 1 polyp, Done. (0.010s)
video 1/1 (5/2238) /content/yolov5/polypp13.avi: 512x640 1 polyp, Done. (0.012s)

video 1/1 (2234/2238) /content/yolov5/polypp13.avi: 512x640 Done. (0.010s)
video 1/1 (2235/2238) /content/yolov5/polypp13.avi: 512x640 Done. (0.011s)
video 1/1 (2236/2238) /content/yolov5/polypp13.avi: 512x640 Done. (0.010s)
video 1/1 (2237/2238) /content/yolov5/polypp13.avi: 512x640 Done. (0.010s)
video 1/1 (2238/2238) /content/yolov5/polypp13.avi: 512x640 Done. (0.010s)
Speed: 0.4ms pre-process, 10.5ms inference, 0.7ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp
```

Figure 4.10: Polyp detection on a video dataset using YOLOv5 model

The figure 4.10 shows the code execution when performing the detection on the video file from colonoscopic examination using the trained YOLOv5 model. The output during the execution processes the video frame by frame. The video consists of 2238 image frames and the model considers each image frame as input data and performs the polyp detection task on it. The detection result is stored as a video containing the highlighted bounding box details for polyp object found in each image frame throughout the video. The video was processed at a speed of 86.20 FPS which is decent for real time processing.

Method	Backbone	AP	IoU	AP25	AP50	AP75	FPS
YOLOv4 [30]	Darknet53, CSP	0.8513	0.8025	0.9123	0.8234	0.7594	48.00
ColonSegNet [73]		0.8000	0.8100	0.9000	0.8166	0.6706	180.00
YOLOv5 [75]	CSPDarknet			0.906	0.903	0.888	111.11
YOLOR [130]	Shared backbone			0.868	0.879	0.682	69.44
PP-YOLO [84]	ConvNet				0.8654		40.900

Table 4.7: Comparison result on the polyp detection and localisation task on the Kvasir-SEG [74] dataset.

4.8 Comparison with earlier benchmark detection models

The results in the table 4.7 indicate that the performance metric result for AP50 is higher for all three models YOLOv5, YOLOR and PP-YOLO when compared to YOLOv4 and ColonSegNet methods. Also, the obtained result for AP75 value for YOLOv5 is highest among all and for YOLOR the value for AP75 is also greater than ColonSegNet method. However, the ranking of the YOLOR model is lowest for value AP25 among all and AP25 value for YOLOv5 is 0.906 which is slightly higher than ColonSegNet 0.900 but lesser than that of YOLOv4 which is 0.9123. With regards to Frames Per Second FPS, the models YOLOv5 having FPS=111.11, YOLOR HAVING FPS=69.444 did fairly well when compared to YOLOv4 FPS=48.00. However, in terms of speed for processing it is significantly lower when compared to ColonSegNet FPS=180.00 which is nearly triple. Therefore, there is still potential future scope for improvement for further improving these metrics and increasing the efficiency in terms of processing speed.

4.9 Summary

The experimental results from three models YOLOv5, YOLOR and PP-YOLO with polyp datasets were enlisted. In the next chapter, a discussion about the results is presented.

Chapter 5

Discussion

5.1 General discussion

Although all of the three models, have default augmentations in place, but still the models could also be trained and evaluated with additional data augmentation techniques with manual settings to see the impact on the training and evaluation results. It would be interesting to have the models could be trained further with running hyperparameter evolution using genetic algorithm as this was not achieved in this thesis due to huge training time required nearly in days.Few more experiments that could be performed for trial and testing is to implement frozen layer concept in the neural networks for transfer learning where the existing model could be retrained on a new dataset without training the entire network. For this, some of the initial weights are frozen and the remaining weights are utilized to compute loss and updated by optimizer. This method therefore requires fewer resources when compared to normal training which result in quicker training times at the cost of slight decrease in the final trained accuracy.

For training the model with convolutional neural network, the image resolution size plays a crucial role. The resizing of all the images in a dataset to a particular image resolution can have an impact on the time required for training and performance of the model. In cases where the images with the bigger size are downscaled, CNN finds it difficult to learn the features necessary for detection and classification since pixels containing key features are reduced. In scenarios where the smaller images when upscaled are padded

with zeroes, the CNN needs to learn the padded pixels that are of no use for image detection or classification task. In addition, bigger images slow the training process and need more memory [104].

5.2 Research objectives

The overall detection performance for all the three models after running the detection model on the test dataset is presented in the tables. The performance evaluation is done with the metrics specified in section 3.6. The goal of achieving the nearly accurate detection predictions to some extent is fulfilled. However, there is still room for improvement in terms of getting higher precision and recall values. As seen in the table, the speed of the detection is lower compared to the earlier benchmark model YOLOv4. To solve this problem, the model training needs to be optimized by modifications in the layers of the neural network while achieving the same level of accuracy. The efficient execution for detecting the polyp objects in the real time images is achieved however the working model for detection in real time videos could not be achieved due to lack of time. The total time taken for detection for 200 test images took seconds for YOLOv5, seconds for YOLOR and seconds for PP-YOLO. The highest rate of FPS was achieved for models that is significantly above the standard value of 25-30 FPS.

The trained model could be easily extended to setting up the environment required for detecting multi-class polyp objects and classifying them along with localization. Only thing that is required is collecting the dataset having proper annotations for bounding boxes for each type of polyp object and further correctly preparing the dataset in the acceptable format required for the models.

5.3 Challenges

This sections gives a spotlight on the actual challenges that were faced in the complete process of running through the experiments. All the experiments were performed using the Google Colaboratory. It is online platform hosting jupyter notebook service where a code in python can be written and executed

in the browser without any setup required for use. It is typically suited for for data analysis and machine learning. However, there are some limitations on the resources and usage limits in Google Colab. The Google Colab instance gets automatically disconnected when the instance remains idle for 90 minutes or the maximum time of the instance where it can remain connected to GPU is 12 hrs. Therefore, in this thesis work, it happened multiple times that the model training got disconnected and stopped in the middle of the execution due to this issue. This led to the rework of again iterating the process of executing the code, preparing of the data and then starting the training process. To avoid this issue, a possible solution was searched on the internet for preventing Google Colab from disconnecting. The following method was found from stackoverflow page [120]. Press Ctrl + Shift + i in the notebook page for opening the inspector view. Navigate to Console, and write the below program statements and hi enter button. This would continue to keep on clicking on the page and help in preventing from disconnecting.

```
function ClickConnect(){
console.log("Working");
document.querySelector("colab-toolbar
-button#connect").click()
}
setInterval(ClickConnect,60000)
```

Another issue that occurred when running the training execution was the below runtime error stating that the all the tensors were not found on the same device.

```
RuntimeError: Expected all tensors to be on the same
device, but found at least two devices, cuda:0 and cpu!
```

For mitigating this error, several articles on the internet were read, but did not work. Finally, the factory reset runtime option was chosen and the execution was restarted again and the error disappeared.

Another major problem when running the training execution was larger batch size, the below fatal error occurred indicating segmentation fault by operation system. This is due to lack of memory space available to process the images in larger batch size. In such cases the batch sizes had to be

reduced which then resulted in longer training execution times.

```
FatalError: ‘Segmentation fault‘ is detected by the
operating system.
```

Wandb is an interactive central dashboard for visualizing and tracking the results involving hyperparameters, predictions and system and performance metrics that enables the real time comparison of models [11]. During one of the several experiments, the following error occured related to wandb.

```
wandb.errors.Error: You must call wandb.init() before
wandb.log()
```

For fixing this issue, several alternatives were tried like disconnecting the wandb login from the Google Colab notebook instance or doing a restart runtime, but the issue did not resolve. In the end, factory resent runtime was performed on the notebook instance which fixed the problem.

In the experimentation for training the PP-YOLO model, where the‘ snapshot of the model trained was saved at every epoch, it resulted in an OSerror in the GPU device as below.

```
OSError: [Errno 28] No space left on device
```

Since lot of disk space was used up due to save of the trained model weights and parameters, it was decided to save the values after every 5 epochs instead of 1 epoch. In addition, this kind of error indicates that the model training should have been carried out on more powerful hardware.

5.4 Lessons learnt

This was first of its kind of experience to work on the thesis involving deep neural networks in the field of computer vision. Therefore, this section emphasizes the key lessons learnt especially while working with deep neural networks. One of the key lessons learnt is that the training of the models using deep neural networks require somewhere between 2 to 12 hrs. Therefore, it is of utmost importance that the training of the models should be scheduled during the daytime and not during the nights. This is to allow to monitor the training execution intermittently and check that the training of the models is still in progress and has not ran into issues. Another thing

to note is that the availability of the GPU resources allocated in Google Colab depends on the previously used resources. The more efficient and faster running GPUs are allocated to the users to have not used them frequently in the past. Therefore, another lesson learnt is that when the code execution does not take place and the code is in editing mode then the GPU session should be manually disconnected. Similarly, when the training or testing execution is finished, then also the GPU session should be terminated. Lastly, the results obtained from different experiments should be noted down simultaneously as later it really becomes a tedious job to manage and document the results.For saving the execution results of train, test or detect into google drive storage, one requires larger available space available in Google Drive because the weights file from the execution output is normally bigger in size. Therefore, there should be sufficient space availability and timely cleanup of data in google drive for storing the result files. Another lesson learnt is if the entire work had to be started again from the beginning, there are few things that could have been taken into consideration. Due to time constraint, the models could not trained and tested on large datasets. Training of the model execution in deep learning requires is time consuming. Therefore, the models could be trained for more datasets for learning more image features. Also, the number of epochs for training could have been increased from 300 to a larger value to check if there is any improvement when increasing the number of epoch value.

Chapter 6

Conclusions

This chapter provides a recap of the work that was done in this thesis for addressing the problem statement discussed in section 1.2. Further, it describes the main contributions and what further enhancements could be done as part of future work to achieve better results.

6.1 Summary

This thesis presents the work and its related experiences with implementation of recent deep learning models for object detection and localization for automatic analysis of medical images from GI tract examinations in order to detect and localize the polyps in the images. The aim of the research work in this thesis was to evaluate and compare the performance results with the earlier state-of-the-art models for polyp object detection. The newly released deep learning models YOLOv5, YOLOR and PP-YOLO in YOLO family of models that have shown improved performance when compared to existing benchmark models on MS-COCO dataset. The goal was that the model implementations could potentially demonstrate better performance results in comparison to earlier researched models for polyp object detection and serve as an automatic polyp detection system.
In conclusion, the objection detection system using YOLOv5, YOLOR and PP-YOLO can be utilized for polyp detection in GI tract medical image diagnosis. The models achieved better training accuracy as compared to the earlier YOLOv4 model. However the detection accuracy on the unseen test-

ing dataset for the models was satisfactory. This suggests that there is still scope for improvements.

6.2 Main contributions

In this thesis work, we have shown that the newer versions of detection models in the YOLO family of models based on deep learning could be utilized effectively and easily for performing polyp detection and localization task. The method YOLOv5 and PP-YOLO achieved fairly good detection accuracy with precision of 0.781 and 0.885 respectively for unseen test dataset. However, the detection accuracy for YOLOR, PP-YOLO model was not satisfactory having precision value of 0.442 despite having good training and evaluation results. The speed of detection for all the three models was adequate with YOLOv5 [75] having FPS=106.38, YOLOR [130] having FPS=76.62 and PP-YOLO [84] having FPS=45.04. The YOLOv5 model achieved striking FPS value of 121.96. The mean average precision values for the models also showed good results having 0.721, 0.628 and 0.888 for YOLOv5, YOLOR and PP-YOLO models. The main contributions in this thesis comprises of researching and configuring a system through actual implementation of existing models for our polyp application scenario which is capable of detecting and localizing the polyp images in the GI medical images with fairly good accuracy and detection speed. This further training, evaluation and then testing of the model on larger dataset containing both images and videos is highly recommended to for generalization of model. The models can be easily extended to built the complete automated polyp detection, localization as well as classification of polyps in GI tract.

6.3 Future work

For future work, the models should be retrained with the obtained best model weights from these models YOLOV5 [75],YOLOR [130] and PP-YOLO [84] with more input images with bounding box annotations for training.The model prediction could output better results through learning from vast number of images. This work can be further extended to particularly detect,

localize and also classify specific type of the polyps:Serrated, Adenomatous, Inflammatory as discussed in section 2.3 using multi-class dataset. This would help to diagnose multiple type of diseases that occur in GI tract. These models could be extended to use video dataset as input and view the real time detection results. The training, validation and testing with real time video dataset would be interesting to check whether the generalization of weights take place for predictions on totally newer and unseen data. The models could be further trained with additional dataset containing images and video frames with their corresponding annotations containing the Kvasir-Instrument [72] for colonoscopic examinations. This would aid the models to distinguish precisely between the instrument and polyps to reduce the occurrences of false positives. Another aspect which needs to be considered in case of the medical images and videos are that because of the improper lighting conditions, the quality and clarity in the dataset is poor. Therefore, such image frames in the dataset should be identified and pre-processed with proper augmentation in order to enhance the model training for further generalization of the model.

Bibliography

[1] Anatomy of the Digestive System | Saint Lukes Health System.

[2] Automatic Log Analysis using Deep Learning and AI.

[3] Capsule endoscopy - Mayo Clinic.

[4] Colorectal Cancer Basics. *Mayo Clinic Health system.*

[5] File:2402 Layers of the Gastrointestinal Tract.jpg - Wikipedia.

[6] Flexible sigmoidoscopy - Mayo Clinic.

[7] A gentle introduction to computer vision.

[8] Paddle detectin.

[9] Sporadic (Nonhereditary) Colorectal Cancer.

[10] VetFolio.

[11] What is wandb.

[12] World Health Organization - International Agency for Research on Cancer. Estimated cancer incidence, mortality and prevalence worldwide in 2012.

[13] Your Digestive system and how it works. *U.S. Department of Health and Human Services, National Institue of Diabetes and Digestive and Kidneys diseases.*

[14] Bowel cancer | cancer research uk. 2007.

[15] Globocan 2018 estimated cancer incidence, mortality and prevalence worldwide in 2018, 2018.

[16] Colorectal cancer incidence in female age 0-85+ in the nordic countries (1957-2016), 2019.

[17] Leufkens A, van Oijen M, Vleggaar F, and Siersema P. Factors influencing the miss rate of polyps in a back-to-back colonoscopy study, 2012.

[18] A. Di Ieva, M. Tam, M. Tschabitscher, and M. D. Cusimano. A journey into the technical evolution of neuroendoscopy. *World Neurosurg.*, 82 no. 6, pp. e777-e789, Dec 2014.

[19] Alexander P Abadir, Mohammed Fahad Ali, William Karnes, and Jason B Samarasena. Artificial intelligence in gastrointestinal endoscopy. *Clin. Endosc.*, 53(2):132–141, March 2020.

[20] Carina Albuquerque, Leonardo Vanneschi, Roberto Henriques, Mauro Castelli, Vanda Povoa, Rita Fior, and Nickolas Papanikolaou. Object detection for automatic cancer cell counting in zebrafish xenografts. *PLOS ONE*, 16:e0260609, 11 2021.

[21] Sharib Ali, Felix Zhou, Barbara Braden, Adam Bailey, Suhui Yang, Guanju Cheng, Pengyi Zhang, Xiaoqiong Li, Maxime Kayser, Roger D Soberanis-Mukul, Shadi Albarqouni, Xiaokang Wang, Chunqing Wang, Seiryo Watanabe, Ilkay Oksuz, Qingtian Ning, Shufan Yang, Mohammad Azam Khan, Xiaohong W Gao, Stefano Realdon, Maxim Loshchenov, Julia A Schnabel, James E East, Georges Wagnieres, Victor B Loschenov, Enrico Grisan, Christian Daul, Walter Blondel, and Jens Rittscher. An objective comparison of detection and segmentation algorithms for artefacts in clinical endoscopy. *Sci. Rep.*, 10(1):2748, 2020.

[22] American Cancer Society . Cancer facts & figures 2020. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2020/cancer-facts-and-figures-2020.pdf.

[23] American Cancer Society . Colorectal cancer screening guidelines. https://www.cancer.org/health-care-professionals/american-cancer-society-prevention-early-detection-guidelines/

colorectal-cancer-screening-guidelines.html#:~:text=The%20American%20Cancer%20Society%202018,personal%20preferences%20and%20test%20availability.

[24] American Cancer Society . Colorectal cancer screening guidelines. https://www.mayoclinic.org/tests-procedures/colonoscopy/about/pac-20393569.

[25] Lihi Gur Arie. The practical guide for object detection with yolov5 algorithm. https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843.

[26] Kay A. Ball. The evolution of endoscopy., 1997.

[27] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures, 2012.

[28] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sanchez, Bogdan J Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjorn Rustad, Ilangko Balasingham, Konstantin Pogorelov, Sungbin Choi, Quentin Debard, Lena Maier-Hein, Stefanie Speidel, Danail Stoyanov, Patrick Brandao, Henry Cordova, Cristina Sanchez-Montes, Suryakanth R Gurudu, Gloria Fernandez-Esparrach, Xavier Dray, Jianming Liang, and Aymeric Histace. Comparative validation of polyp detection methods in video colonoscopy: Results from the MICCAI 2015 endoscopic vision challenge. *IEEE Trans. Med. Imaging*, 36(6):1231–1249, June 2017.

[29] Hanoi University of Science BK.AI, Technology incorporation with Institute of Gastroenterology, and Hepatology (IGH) Vietnam. Bkai-igh neopolyp a dataset for colonoscopy polyp segmentation and neoplasm characterization.

[30] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[31] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[32] Pox CP Brenner H, Kloor M. Colorectal cancer. 2014.

[33] Jason Brownlee. Difference between a batch and an epoch in a neural network, 2018.

[34] Jason Brownlee. A gentle introduction to object recognition with deep learning, 2019.

[35] Jason Brownlee. How to control the stability of training neural networks with the batch size, 2020.

[36] Cancer.Net. Colorectal cancer: Risk factors and prevention. https://www.cancer.net/cancer-types/colorectal-cancer/risk-factors-and-prevention, 2021.

[37] Cancer.Net. Colorectal cancer: Stages. https://www.cancer.net/cancer-types/colorectal-cancer/stages, 2021.

[38] Cancer.Net. Colorectal cancer: Statistics. https://www.cancer.net/cancer-types/colorectal-cancer/statistics#:~:text=Worldwide%2C%20colorectal%20cancer%20is%20the%20second%20leading%20cause%20of%20cancer,it%20can%20often%20be%20cured., 2022.

[39] Cancer.Net. Colorectal cancer: Statistics approved by the cancer.net editorial board. https://www.cancer.net/cancer-types/colorectal-cancer/statistics, 2022.

[40] Cleveland Clinic. Gastrointestinal diseases. https://my.clevelandclinic.org/health/articles/7040-gastrointestinal-diseases.

[41] Mayo Clinic. Flexible sigmoidoscopy overview. https://www.mayoclinic.org/tests-procedures/flexible-sigmoidoscopy/about/pac-20394189.

[42] Mayo Clinic. Upper endoscopy overview. https://www.mayoclinic.org/tests-procedures/endoscopy/about/pac-20395197.

[43] Mayo Clinic. Colon polyps, 2021.

[44] Colorectal Cancer Prevention Network at the University of South Carolina . Statistics. https://www.crcfacts.com/statistics.html#:~:text=The%20colorectal%20cancer%20death%20rate,screening%2C%20and%20improvements%20in%20treatment.

[45] Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, Paul R. Young, and Peter J. Denning. Computing as a discipline. *Computer*, 22(2):63–70, feb 1989.

[46] Wikimedia Commons. File:typical cnn.png — wikimedia commons, the free media repository, 2022. [Online; accessed 9-May-2022].

[47] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[48] danielzhangau. Paddle detection how to prepare training data.

[49] Italy Darian Frajberg Scientific seminar at Politecnico di Milano Como. Introduction to the artificial intelligence and computer vision revolution, 2017.

[50] Thomas de Lange, Pål Halvorsen, and Michael Riegler. Methodology to develop machine learning algorithms to improve performance in gastrointestinal endoscopy. *World J. Gastroenterol.*, 24(45):5057–5062, December 2018.

[51] Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[52] Dorling Kindersley Limited. Endoscopy. British Medical Association Complete Family Health Encyclopedia, 1990.

[53] Brad Dwyer. Advanced augmentations in roboflow, 2020.

[54] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Jun 2010.

[55] Ross Girshick. Fast r-cnn, 2015.

[56] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

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

[58] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2017.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[60] D. Heresbach, T. Barrioz, M. G. Lapalus, D. Coumaros, P. Bauret, P. Potier, D. Sautereau, C. Boustière, J. C. Grimaud, C. Barthélémy, J. Sée, I. Serraj, P. N. D'Halluin, B. Branger, T. Ponchon, and the Société Française d'Endoscopie Digestive (SFED). Miss rate for colorectal neoplastic polyps: a prospective multicenter study of back-to-back video colonoscopies, 2008.

[61] David G Hewett, Charles J Kahi, and Douglas K Rex. Efficacy and effectiveness of colonoscopy: how do we bridge the gap? *Gastrointest. Endosc. Clin. N. Am.*, 20(4):673–684, October 2010.

[62] Ø. Holme, M. Bretthauer, A. Fretheim, J. Odgaard-Jensen, and G. Hoff. Flexible sigmoidoscopy versus faecal occult blood testing for colorectal cancer screening in asymptomatic individuals, 2013.

[63] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[64] 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, 2017.

[65] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[66] Xin Huang, Xinxin Wang, Wenyu Lv, Xiaying Bai, Xiang Long, Kaipeng Deng, Qingqing Dang, Shumin Han, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, Yanjun Ma, and Osamu Yoshie. Pp-yolov2: A practical object detector, 2021.

[67] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size, 2016.

[68] Dana-Farber Cancer Institute. Understanding upper endoscopy and colonoscopy. https://www.dana-farber.org/health-library/articles/understanding-upper-endoscopy-and-colonoscopy/.

[69] World Cancer Research Fund International. Worldwide cancer data, 2020.

[70] Debesh Jha. 2020 mediaeval medico - polyp segmentation. https://github.com/DebeshJha/2020-MediaEval-Medico-polyp-segmentation, 2020.

[71] Debesh Jha. Machine learning-based classification, detection, and segmentation of medical images. 2022.

[72] Debesh Jha, Sharib Ali, Krister Emanuelsen, Steven A. Hicks, Vajira Thambawita, Enrique Garcia-Ceja, Michael A. Riegler, Thomas de Lange, Peter T. Schmidt, Håvard D. Johansen, Dag Johansen, and Pål Halvorsen. Kvasir-instrument: Diagnostic and therapeutic tool segmentation dataset in gastrointestinal endoscopy. In *MultiMedia Modeling*, pages 218–229, 2021.

[73] Debesh Jha, Sharib Ali, Nikhil Kumar Tomar, Håvard D Johansen, Dag Johansen, Jens Rittscher, Michael A Riegler, and Pål Halvorsen. Real-time polyp detection, localization and segmentation in colonoscopy using deep learning. *Ieee Access*, 9:40496–40510, 2021.

[74] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *International Conference on Multimedia Modeling*, pages 451–462. Springer, 2020.

[75] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, February 2022.

[76] Ritesh Kanjee. Is yolor better and faster than yolov4?, 2021.

[77] Kjellmo, Ase, and Anders Drolsum. Diagnostikk og stadieinndeling av kolorektal kreft (Diagnosis and staging of colorectal cancer). *Tidsskrift for den Norske laegeforening : tidsskrift for praktisk medicin, ny raekke*, 127(21), 2007.

[78] I. K. "Larsen and F." Bray. Trends in colorectal cancer incidence in norway 1962-2006 an interpretation of the temporal patterns by anatomic subsite, 2010.

[79] Ji Young Lee, Jinhoon Jeong, Eun Mi Song, Chunae Ha, Hyo Jeong Lee, Ja Eun Koo, Dong-Hoon Yang, Namkug Kim, and Jeong-Sik Byeon. Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets. *Scientific Reports*, 10(1):8379, May 2020.

[80] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[81] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[82] Ming Liu, Jue Jiang, and Zenan Wang. Colonic polyp detection in endoscopic videos with single shot detection based deep convolutional neural network. *IEEE Access*, 7:75058–75066, 2019.

[83] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing, 2016.

[84] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, and Shilei Wen. Pp-yolo: An effective and efficient implementation of object detector, 2020.

[85] Arnold M, Sierra MS, Laversanne M, Soerjomataram I, Jemal, and Bray F. Global patterns and trends in colorectal cancer incidence and mortality, 2017.

[86] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design, 2018.

[87] Gaurav Maindola. A brief history of yolo object detection models from yolov1 to yolov5, 2021.

[88] Gaurav Maindola. Introduction to yolov5 object detection with tutorial, 2021.

[89] Culjat Martin, Rahul Singh, and Hua Lee. Medical devices surgical and image-guided technologies. https://vdoc.pub/documents/medical-devices-surgical-and-image-guided-technologies-4ivno04i2n50, 2012.

[90] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks, 2018.

[91] Mayo Clinic. Patient care and health information > tests and procedures > colonoscopy. https://www.mayoclinic.org/tests-procedures/colonoscopy/about/pac-20393569.

[92] Gideon Mendels. Selecting the right weight initialization for your deep neural network, 2019.

[93] Appsilon DataScience Michal Maj. Object detection and image classification with yolo.

[94] Mayank Mishra. Convolutional neural networks, explained. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939, 2020.

[95] Ahmed Mohammed, Sule Yildirim, Ivar Farup, Marius Pedersen, and Øistein Hovde. Y-net: A deep convolutional neural network for polyp detection. 2018.

[96] Ken Namikawa, Toshiaki Hirasawa, Toshiyuki Yoshio, Junko Fujisaki, Tsuyoshi Ozawa, Soichiro Ishihara, Tomonori Aoki, Atsuo Yamada, Kazuhiko Koike, Hideo Suzuki, and Tomohiro Tada. Utilizing artificial intelligence in endoscopy: a clinician's guide. *Expert Review of Gastroenterology & Hepatology*, 14(8):689–706, 2020. PMID: 32500760.

[97] Nancy Bazilchuk. More colorectal cancer in norway than elsewhere in europe. https://sciencenorway.no/cancer-forskningno-norway/more-colorectal-cancer-in-norway-than-elsewhere-in-europe/1424891, 2015.

[98] National Cancer Institute Surveillance, Epidemiology, and End Results (SEER) Program. Cancer Stat Facts: Colorectal Cancer. https://sciencenorway.no/cancer-forskningno-norway/more-colorectal-cancer-in-norway-than-elsewhere-in-europe/1424891.

[99] Tim Newman. Endoscopy: What to know. https://www.medicalnewstoday.com/articles/153737#preparation, 2022.

[100] NHS. Endoscopy. https://www.nhs.uk/conditions/endoscopy/.

[101] University of Michigan Health Michigan Medicine. Colon and rectal polyps.

[102] World Health Organization. Cancer, 2022.

[103] Rahul Pannala, Kumar Krishnan, Joshua Melson, Mansour Parsi, Allison Schulman, Shelby Sullivan, Guru Trikudanathan, Arvind Trindade, Rabindra Watson, John Maple, and David Lichtenstein. Artificial intelligence in gastrointestinal endoscopy. *VideoGIE*, 5:598–613, 12 2020.

[104] Aravind Ramalingam. How to pick the optimal image size for training convolution neural network?, 2021.

[105] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

[106] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[107] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[108] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

[109] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[110] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[111] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[112] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[113] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[114] Saint Luke's. Lower gi endoscopy. https://www.saintlukeskc.org/health-library/lower-gi-endoscopy.

[115] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018.

[116] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging*, 35(5):1285–1298, May 2016.

[117] Younghak Shin, Hemin Ali Qadir, Lars Aabakken, Jacob Bergsland, and Ilangko Balasingham. Automatic colon polyp detection using region based deep CNN and post learning approaches. *IEEE Access*, 6:40950–40962, 2018.

[118] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[119] The American Cancer Society. Key statistics for colorectal cancer.

[120] stackoverflow. How to prevent google colab from disconnecting?

[121] M. Sukkar, D. Kumar, and J. Sindha. Real-time pedestrians detection by yolov5, 2021.

[122] Shahriar Shakir Sumit, Junzo Watada, Anurava Roy, and DRA Rambli. In object detection deep learning methods, YOLO shows supremum to mask r-CNN. *Journal of Physics: Conference Series*, 1529(4):042086, apr 2020.

[123] SuperAnnotate. Introduction to object detection with deep learning. https://blog.superannotate.com/object-detection-with-deep-learning/?utm_term=&utm_campaign=Annotation+Search+New+Segmented+-+Dynamic&utm_source=adwords&utm_medium=ppc&hsa_acc=7527629942&hsa_cam=15242080204&hsa_grp=135322064288&hsa_ad=561071592633&hsa_src=g&hsa_tgt=dsa-1118336060137&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=

CjwKCAjwur-SBhB6EiwA5sKtjiwSUUI34NCPx9-smfp0L21jiZJeC2IjH8lyt8jHbgLXTb3taw5osRoC
BwE, 2021.

[124] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans. Med. Imaging*, 35(5):1299–1312, May 2016.

[125] Aysegul Takimoglu. Top data augmentation techniques: Ultimate guide for 2022, 2022.

[126] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. 2018.

[127] Hasty visionAI Wiki. Bounding box regression loss. https://wiki.hasty.ai/loss/bounding-box-regression-loss.

[128] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network, 2020.

[129] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. Cspnet: A new backbone that can enhance learning capability of cnn, 2019.

[130] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks, 2021.

[131] MD WebMD, Carol DerSarkissian. Digestive diseases and endoscopy. https://www.webmd.com/digestive-disorders/digestive-diseases-endoscopy, 2021.

[132] Wikipedia contributors. Artificial intelligence — Wikipedia, the free encyclopedia, 2022. [Online; accessed 9-May-2022].

[133] Wikipedia contributors. Deep learning — Wikipedia, the free encyclopedia, 2022. [Online; accessed 9-May-2022].

[134] Wikipedia contributors. Endoscopy — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Endoscopy&oldid=1081665846, 2022. [Online; accessed 4-May-2022].

[135] Wikipedia contributors. Gastrointestinal tract — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Gastrointestinal_tract&oldid=1084717953, 2022. [Online; accessed 3-May-2022].

[136] Wikipedia contributors. Virtual colonoscopy — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Virtual_colonoscopy&oldid=1064933776, 2022. [Online; accessed 4-May-2022].

[137] Code with Aarohi. Yolor on a custom dataset | object detection using yolor, 2022.

[138] WongKingYiu. Yolor implementation of paper - you only learn one representation: Unified network for multiple tasks.

[139] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[140] Yale Medicine. Gastrointestinal cancers overview. https://www.yalemedicine.org/conditions/gastrointestinal-cancers.

[141] Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John Paisley. Pannet: A deep network architecture for pan-sharpening. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1753–1761, 2017.

[142] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark, 2015.

[143] Young Joo Yang and Chang Seok Bang. Application of artificial intelligence in gastroenterology. *World J. Gastroenterol.*, 25(14):1666–1683, April 2019.

[144] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng Ann Heng. Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos. *IEEE Journal of Biomedical and Health Informatics*, 21(1):65–75, 2017.

[145] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices, 2017.

[146] Xu Zhang, Fei Chen, Tao Yu, Jiye An, Zhengxing Huang, Jiquan Liu, Weiling Hu, Liangjing Wang, Huilong Duan, and Jianmin Si. Real-time gastric polyp detection using convolutional neural networks. *PLoS ONE*, 14, 2019.

[147] Yali Zheng, Ruikai Zhang, Ruoxi Yu, Yuqi Jiang, Tony W. C. Mak, Sunny H. Wong, James Y. W. Lau, and Carmen C. Y. Poon. Localisation of colorectal polyps by convolutional neural network features learnt from white light and narrow band endoscopic images of multiple databases. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4142–4145, 2018.

[148] Rongsheng Zhu, Rong Zhang, and Dixiu Xue. Lesion detection of endoscopy images based on convolutional neural network features. In *2015 8th International Congress on Image and Signal Processing (CISP)*, pages 372–376, 2015.

Appendix A

Repository for Model configurations

A.1 Code execution google colab pdf files for YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOR and PP-YOLO

A.2 Code execution .ipyb files for YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOR and PP-YOLO

Appendix B

Model Training results

B.1 Model Training results with Kvasir-SEG base dataset for YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOR and PP-YOLO

B.2 Model Training results with polyp dataset with normal images not having polyps for YOLOv5, YOLOR and PP-YOLO

B.3 Model retraining results with BKAI-IGP NeoSmall-Polyp for YOLOv5, YOLOR and PP-YOLO

Appendix C

Model Testing results

C.1 Model Testing results on polyp-mediai test dataset with images YOLOv5, YOLOR and PP-YOLO containing the images, ground truths and images with polyp detections

C.2 Testing results on video dataset for YOLOv5, YOLOR and PP-YOLO containing the video and detections on the video