# Polyp Detection in Colonoscopy Images

A Report submitted

in partial fulfilment of the requirement for the

degree of

## BACHELOR OF TECHNOLOGY

In

## Computer Science & Engineering

By


Mohd Aamir (2001640100159)
Sudhir Kumar (2001640100267)
Mohd Faraz Akram (2001640100160)
Mohammad Saquib Abbas (2001640100158)
Mohd Ateeb Khan (2001640100157)


Under the supervision of

**Mr. Durgesh Pandey**
**(Assistant Professor ,CSE Department)**


**Pranveer Singh Institute of Technology ,Kanpur**



**Dr. APJ Abdul Kalam Technical University**
**Lucknow**

**May, 2024**

# CERTIFICATE

This is to certify that Project Report entitled "**Polyp Detection in Colonoscopy Images**" which is submitted by **Mohd Aamir, Sudhir Kumar, Mohd Faraz Akram, Mohammad Saquib Abbas, Mohd Ateeb Khan** in partial fulfilment of the requirement for the award of degree **B. Tech**. in Department of **Computer Science and Engineering** of **Pranveer Singh Institute of Technology**, affiliated to **Dr. A.P.J. Abdul Kalam Technical University, Lucknow** is a record of the candidates own work carried out by them under my supervision. The project embodies result of original work and studies carried out by the students themselves and the contents of the project do not form the basis for the award of any other degree to the candidate or to anybody else.

Signature:

Prof. (Dr.) Vishal Nagar
Dean- CSE,
PSIT, Kanpur

Signature:

Mr. Durgesh Pandey
Assistant Professor
CSE Department,
PSIT, Kanpur

Date:

# DECLARATION

We hereby declare that the work presented in this report entitled "**Polyp Detection in Colonoscopy Images**", was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

**Signature**
**Name: Mohd Aamir**
**Roll No.: 2001640100159**

**Signature**
**Name: Sudhir Kumar**
**Roll No.: 2001640100267**

**Signature**
**Name: Mohd Faraz Akram**
**Roll No.: 2001640100160**

**Signature**
**Name: Mohammad Saquib Abbas**
**Roll No.: 2001640100158**

**Signature**
**Name: Mohd Ateeb Khan**
**Roll No.: 2001640100157**

**Date:**

# ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. Final Year. We owe special debt of gratitude to our project supervisor Mr. Durgesh Pandey (Assistant Professor), Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur for his constant support and guidance throughout the course of our work. His sincerely, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavours have seen light of the day.

We also take the opportunity to acknowledge the contribution of Prof.(Dr.) Vishal Nagar, Dean, Computer Science & Engineering, Pranveer Singh Institute of Technology, Kanpur for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature
Name: Mohd Aamir
Roll No.: 2001640100159

Signature
Name: Sudhir Kumar
Roll No.: 2001640100267

Signature
Name: Mohd Faraz Akram
Roll No.: 2001640100160

Signature
Name: Mohammad Saquib Abbas
Roll No.: 2001640100158

Signature
Name: Mohd Ateeb Khan
Roll No.: 2001640100157

# ABSTRACT

Detection and localization methods can help improve colonoscopy procedures and with the help of these methods we can detect polyp with more accuracy and speed. Despite the fact that various approaches have been developed to deal with automatic polyp identification, benchmarking of state-of-the-art methods still remains an open problem. This is because to the growing variety of computer vision technologies that have been studied and may be applied to polyp datasets. Benchmarking innovative approaches (such as YOLO, SSD, Faster RCNN and so on) can help guide the development of automated polyp detection tasks. Current polyp detection approaches based on object detection frameworks need many handcrafted pre- and post-processing procedures, as well as user assistance, which necessitate domain specific knowledge. In this report, we have proposed an automated polyp detection approach using Deformable DETR model which is based on transformer. This detector is then used for the automatic detection of polyp in the image. These images are basically obtained from colonoscopy. Further we have improved our results by changing the loss function and applying some augmentation strategies. The result that we obtained from our proposed approach show that our approach outperforms the state-of-the-art methods.

# TABLE OF CONTENTS

**TITLE** **PAGE NO.**

**DECLARATION…….…… ii**

**CERTIFICATE……iii**

**ABSTRACT……iv**

**ACKNOWLEDGEMENT……v**

**LIST OF TABLES……vi**

**LIST OF FIGURES……vii**

**CHAPTER-1-INTRODUCTION……1**

1.1 Introduction……...…….…1

1.2 Objective……4

1.3 Contributions…….....4

1.4 Organisation of the Report…….5

**CHAPTER-2-RELATED_WORKS……..6**

2.1 Prior Work….....6

2.2 Work done in the phase I ….....7

2.2.1 YOLO……...8

2.2.2 Faster RCNN…….14

2.3 Dataset……..19

**CHAPTER-3-PROPOSED_APPROACH……..21**

3.1 Detection Transformer…… 21

3.1.1 Why DETR?…….24

3.1.2 Architecture of Detection Transformer…….29

3.1.3 Loss Function……30

3.2 Deformable DETR……..32

**CHAPTER-4-EXPERIMENTS AND RESULTS……36**

4.1 Experimental Dataset.....36

4.2 Evaluation Matric.....36

4.3 Experimental Settings.....37

4.4 Encoder-Decoder Layers….....37

4.5 Comparison of Deformable DETR with YOLOv4 Result.....38

4.6 Loss Function Improvement.....41

4.7 Image Augmentation …….....48

4.8 Final Result Comparison……..48

4.9 Test Time Augmentation…….. 49

**CHAPTER-5-CONCLUSION AND FUTURE WORK……..……..…….…….……..53**

**REFERENCES……...54**

## LIST OF FIGURES

**FIGURE** **PAGE NO.**

2.1 Comparison of Different YOLO Versions ……..10

2.2 YOLOv3 Architecture . . . . . .…… . . . . . .11

2.3 YOLOv3+spp Architecture [HW19] . . . . . .…….. .12

2.4 YOLOv4 Architecture [BWL20] . . . . . .……... ..13

2.5 Faster R-CNN Architecture . . . . . .…… .15

2.6 Sample images from Kvasir-SEG dataset: Original Image(1st column) and

Ground truth bounding boxes (3rd column) for selected samples……....20

3.1 Difference between convolution and attention . . ……. . . . . 28

3.2 DETR Architecture [CMS+20] . . . . . .……. 29

3.3 DETR Transformer Architecture [CMS+20] . . . . . .…….. 31

3.4 Deformable DETR Transformer Architecture [ZSL+20] . . . . . .……. 32

4.1 Precision, Recall and IoU (intersect over union) . . . . . .…….. 37

4.2 Graph between AP at IoU threshold 0.5 and different number of layers …….38

4.3 Training and Validation loss curve of Deformable DETR . . . .…….. . . 38

4.4 Results that are good in both YOLOv4 and Deformable DETR . …… . 39

4.5 Results that are not good in YOLOv4 and but good in Deformable DETR . ... . . . 40

4.6 Results that are bad in both YOLOv4 and Deformable DETR . …… . . 41

4.7 GIoU loss: Blue is the predicted bounding box using GIoU loss[ZWL+19] ……..43

4.8 d and c between predicted and ground truth box[ZWL+19] . . . . . …… 44

4.9 DIoU loss: Red is the predicted bounding box using DIoU loss[ZWL+19] ……..44

4.10 regression error sum curves of different loss functions for different iterations[ZWL+19]. . . . . ……..... . 46

4.11 Results that are bad in both YOLOv4 and Deformable DETR with GIoU loss but improved in Deformable DETR with CIoU loss . . ……... . 47

4.12 Results that are bad even in Deformable DETR with CIoU loss . . …… 47

4.13 Results that are not good earlier but after augmentation they becomes recognisable . . . . . ……. . 49

4.14 This figure shows that our model is able to detect small object. . . ……..... 50

4.15 This figure shows that our model is able to detect multiple object as well. . .. . 51

# LIST OF TABLES

**TABLE** **PAGE NO.**

4.1 Result on the polyp detection task on the Kvasir-SEG dataset and best score is highlighted in last 3 columns. ……..……..……..50

4.2 Result on the polyp detection task on the Augmented Kvasir-SEG dataset and best score is highlighted in last 3 columns …….……. …….50

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Colorectal cancer is a type of cancer that begins in the colon or rectum, parts of the large intestine. It's one of the most common cancers worldwide and typically develops from polyps, which are abnormal growths in the lining of the colon or rectum. Not all polyps turn into cancer, but certain types, such as adenomatous polyps, have a higher risk of becoming cancerous over time.

### Risk Factors

Several factors can increase the risk of developing colorectal cancer:

- Age: Most cases occur in people over 50.
- Family History: A family history of colorectal cancer or polyps can increase risk.
- Genetic Mutations: Conditions like Lynch syndrome and familial adenomatous polyposis (FAP) are genetic disorders that elevate the risk.
- Lifestyle Factors: Diets high in red and processed meats, physical inactivity, obesity, smoking, and heavy alcohol use are linked to a higher risk.
- Inflammatory Bowel Disease: Conditions like Crohn's disease and ulcerative colitis can increase risk.

### Symptoms

Common symptoms of colorectal cancer include:

- Changes in bowel habits (diarrhea or constipation)
- Blood in the stool
- Unexplained weight loss
- Abdominal discomfort or pain
- Fatigue and weakness

**Diagnosis**

Diagnosis typically involves several steps:

- Screening Tests: Such as colonoscopy, sigmoidoscopy, and stool tests.
- Imaging Tests: CT scans or MRIs to view the colon and surrounding organs.
- Biopsy: Taking a tissue sample during a colonoscopy for lab analysis.

**Treatment**

Treatment depends on the stage of the cancer and may include:

- Surgery To remove the tumor and nearby lymph nodes.
- Radiation Therapy: Often used for rectal cancer to shrink tumors before surgery.
- Chemotherapy: Uses drugs to kill cancer cells, often used if cancer has spread.
- Targeted Therapy: Uses drugs that target specific cancer cell mechanisms.
- Immunotherapy: Helps the immune system recognize and attack cancer cells.

**Prevention**

Preventative measures include regular screening, maintaining a healthy diet, regular physical activity, limiting alcohol consumption, and avoiding smoking. For those with a high genetic risk, genetic counseling and regular monitoring are advised.

Early detection through screening is crucial, as colorectal cancer is highly treatable when found early.

The mortality rate of Colorectal Cancer is third highest among all cancers worldwide. Only 68% survival rate of colon cancer is reported over five years, and for stomach cancer it is around 44% [AJpt]. One of the most effective ways to reduce CRC-related mortality is to look for and remove pre-cancerous abnormalities. Polyps in the colon are crucial to diagnose because they can develop into CRC (Colorectal Cancer) at a late stage, which can take several years. So, survival is dependent on early recognition of the CRC.

CRC can be prevented by frequently screening the colon and this procedure is known as colonoscopy. Different research studies suggest that there is high quality evidence that both flexible sigmoidoscopy and faecal occult blood testing reduce colorectal cancer mortality when applied as screening tools [Ht]. Colonoscopy is a medical treatment in which an endoscopist uses a flexible endoscope to check or inspect the colon. A long and flexible tube called colonoscope is inserted inside the colon during the while process. The doctor can see the inside of the colon by using 1 a tiny video camera at the tube's tip. Camera transmits images continuously that can be visible to the monitor, by using which the doctor examines the colon. The doctor can also use the channel to insert devices to take tissue samples (biopsies) or remove polyps or other problematic tissues. Aleast 40 to 50 minutes are required to complete the colonoscopy process. It is widely regarded as the most effective diagnostic tool for colon inspection in the early stages of CRC.

Like all types of cancer, colon cancer happens when cells grow and divide uncontrollably. All cells in your body are constantly growing, dividing and dying. That's how your body remains healthy and working as it should. In colon cancer, cells lining your colon and rectum keep growing and dividing even when they're supposed to die. These cancerous cells may come from polyps in your colon.

Medical researchers aren't sure why some people develop precancerous colon polyps that become colon cancer. They do know certain risk factors increase people's chances of developing precancerous polyps and colon cancer.

Those risk factors include certain medical conditions, including inherited conditions, and lifestyle choices. Having one or more risk factors for colon cancer doesn't mean you'll develop the condition. It just means you have increased risk. Understanding risk factors may help you decide if you should talk to a healthcare provider about your risk of developing colon (colorectal) cancer.

Polyps are abnormal or we can say extra growths on the colon's or rectum's inner lining. They can arise everywhere in the GI (gastrointestinal) system, but they're most common in the

colorectal area, and they're typically thought of as a precursor or pre-cancerous stage of CRC [JSR+19b], [HR01]. There are two types of colorectal polyps: neoplastic and non-neoplastic colorectal polyps. Non-neoplastic polyps are further sub-categorised into Hyperplastic, inflammatory and hamartomatous polyps. These polyps are not cancerous and are not dangerous. Neoplastic polyps are further sub-categorised into Adenomas and serrated polyps. These polyps have the potential to become cancerous. There are three groups in which size of colorectal polyps is categorized: tiny (0.5cm), small (0.6cm to 0.9cm), and progressed (big) (1.0cm) [J.ul]. Larger polyps can usually be found and removed.

These are the factors that can tell if the polyp can contain cancer or increase the risk of developing CRC as follows:

- If polyp larger than 1cm is found.
- If more than 3 polyps are found.
- If the polyp has dysplasia, it should be removed. Dysplasia is another precancerous condition. It indicates that the cells inside a polyp or the lining of the colon or rectum appear abnormal, but they haven't turned cancerous.

### 1.2 Objective

CRC is the fourth most frequently occurring cancer worldwide. If extra growth or abnormal growth is there on the colon surface then we can say it is a polyp which can be cancerous or has a potential to turn into the cancer over time. Our work aims at detecting these polyps by using deep learning techniques that could assist in early cancer diagnosis.

### 1.3 Contributions

**First**, to the best of our knowledge, we are the first to apply the multi-scale Transformer based technique for polyp detection. The methods adopted so far for polyp detection have performed well using deep learning models but their efficiency is still a challenge and they are likely to miss small-sized polyps. In this work, we have investigated transfer learning for polyp

detection using a Detection Transformer, which has achieved good performance on small polyps as well.

**Second**, we performed several experiments to find the best number of encoder and decoder layers in transformer.

**Third**, we have introduced a Loss function and replace the old loss function by new one.

**Fourth**, due to less data we have explored different types of augmentations such as Rotation, shearing, flipping, scaling and gaussian blur for polyp detection and also we have introduces TTA (Test Time Augmentation) to enhance the model performance.

Finally, based on the large number of experiments with polyp images we can say that our object detection model shows improvement over other recent CNN based studies or state-ofthe-art methods in colonoscopy image dataset.

### 1.4 Organization of the Report

This chapter provides a basic overview of the CRC, Colonoscopy and polyps. The organisation of the remaining chapters is given as follows: In Chapter 2, we have provided the course of related work and progress made in this area and also discuss about the state-ofthe-art methods and datasets. In Chapter 3, we have provided the architecture of actual method that we have used (i.e. Deformable Detection Transformer) for the experiment primarily focuses on working of DETR (i.e. Detection Transformer). In Chapter 4, we have discussed about evaluation metrics that we have used, experimental setups and the comparison of results for all experiments. In Chapter 5, we have concluded our work that we have performed along with the future work.

# CHAPTER 2

# RELATED WORKS

## 2.1 Prior-Work

Over the last two decades, polyp identification has been a popular area in research, with considerable work devoted into building efficient and algorithmic techniques. Earlier research focused on polyp colour and texture, with feature learning based on handcrafted descriptions [KSep], [AWP+09]. More recently, CNN based methods have gotten a lot of attention [TSG+16] [SHeb], and they've been the go-to approach for those competing in public competitions [BTS+17], [Alian].

Wang et al.[WYpr] developed algorithms for quick polyp edge and polyp shot detection, as well as a polyp alert software system.

Shin et al.[SQA+18] have done automatic polyp detection in colonoscopy videos and images have by using region-based CNN. They used Inception ResNet as a transfer learning strategy and post-processing approaches for reliable polyp diagnosis in colonoscopy.

Shin et al.[SQB18] employed a generative adversarial network [GPAM+14], demonstrating that while the generated polyp pictures are not qualitatively realistic, they can help with detection.

Lee et al.[Leepr] used YOLOv2 [RDGF15], [RF16] for the creation of polyp detection algorithm. The method generated great sensitivity and performance that was close to real-time.

Yamada et al.[Yampt] developed an artificial intelligence system that can detect the sign of CRC with high sensitivity and specificity during a colonoscopy. They stated that their technology could help endoscopists to spot abnormalities in real time, allowing for early illness diagnosis.

Shin et al.[QSS+21] developed a systen called F-CNN (fully convolutional neural networks) for real time poly detection.

Jha et al.[JAT+21] for the identification and localization challenge, he developed a method called ColonSegNet that achieved a better average precision of 0.8080 and a mean IoU of 0.8110, as well as a great frame rate around 180 fps. He also performed experiments on stateof-the-art methods such as RetinaNet, Faster-RCNN, YOLOv3 [RF16] and YOLOv4[BWL20] on Kawasir-SEG dataset.

Shen et al.[SLZ21] proposed a model named COTR (i.e. convolution in transformer). He simply used a convolution layer interleaved with transformer encoder layers for feature encoding and recalibration and the remaining architecture remains same as DETR (i.e. Detection transformer). COTR successfully improved the slow convergence of DETR.

### 2.2 Work done in the phase I

Object class prediction and regress bounding boxes for localisation are the goals of detection algorithms. The baseline detection and localization methods utilised for the automated detection of polyps in the Kvasir-SEG[JSR+19a] dataset are described in this section.

### DETECTION AND LOCALIZATION BASELINE METHODS

Input, backbone, neck, and head are among the detection mechanisms. We've used images as a source of data. VGG16, ResNet50, ResNext-101, and Darknet are examples of CNN architectures that can be used as the backbone. FPN, PANet and Bi-FPN are called neck which is the subset of backbone network. The prediction boxes are handled by the head, which can be a one-stage detector for dense prediction (e.g., YOLO, RPN, and RetinaNet).

### 2.2.1 YOLO

YOLO is a real-time object identification technique that use neural networks. The speed of YOLO is very fast and precision is also good, because of these reasons thsi algorithm is widely popular. Many applications was developed by YOLO algorithm. YOLO is a far quicker algorithm than its competitors, at speeds up to 45 frames per second.

The YOLO approach makes use of CNN for completion of the object detection task in real time.

In just one forward pass across the network this is able to detect objects..

This indicates that a single algorithm run is used to anticipate the entire image. It has an ability to predict multiple bounding boxes and their corresponding class scores at the same time by using CNN.

YOLO is a family of object detection models designed for real-time processing. Unlike traditional methods, YOLO frames object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. This unique approach makes YOLO extremely fast and efficient.

**Versions of YOLO**

1. **YOLOv1:**

- Architecture: Single convolutional network.
- Speed: Real-time object detection.
- Limitations: Struggles with small objects and multiple objects close to each other.

**2. YOLOv2 (YOLO9000):**

- Improvements: Batch normalization, high-resolution classifier, and anchor boxes.
- Capabilities: Detects over 9000 object categories.

**3. YOLOv3:**

- Architecture: Darknet-53 backbone.

- Features: Multi-scale predictions, residual blocks, and improved performance on small objects.

**4. YOLOv4:**

- Enhancements: CSPDarknet53 backbone, PANet path aggregation, and Mish activation.
- Performance: Higher accuracy and speed compared to YOLOv3.

**5. YOLOv5:**

- Developed by Ultralytics: Not by original authors.
- User-Friendly: Simplified implementation with pre-trained models.
- Scalability: Various model sizes to balance speed and accuracy.

**6. YOLOv6:**

- Industry Focused: Optimized for practical applications.
- Performance: Enhanced deployment efficiency.

**7. YOLOv7:**

- Latest Improvements: New architecture and training strategies.
- State-of-the-Art: Leading in speed and accuracy at the time of release.

**8. YOLOv8:**

- Modern Enhancements: Latest techniques in computer vision.
- Integration: Easy to use with modern AI workflows.

**Applications of YOLO**

1. Healthcare:

Medical Imaging: Detecting polyps in colonoscopy images, tumor detection in radiology.

Benefits: Speed and accuracy enable real-time analysis during procedures.

2. Security:

Surveillance Systems: Real-time monitoring and threat detection.

Advantage: Fast response times and high accuracy.

3. Autonomous Vehicles:

Object Detection: Identifying pedestrians, vehicles, and obstacles.

Critical for Safety: Real-time detection is crucial for safe navigation.

**Detailed Comparison of YOLO Versions**

| Version | Architecture | Key Features | Performance | Use Cases |
|---|---|---|---|---|
| YOLOv1 | Single CNN | Real-time detection | Fast but less accurate for small objects | Basic object detection |
| YOLOv2 | YOLO9000 | Batch normalization, anchor boxes | Better accuracy, detects 9000+ categories | General object detection |
| YOLOv3 | Darknet-53 | Multi-scale predictions, residual blocks | Improved small object detection | Versatile use cases |
| YOLOv4 | CSPDarknet53 | PANet, Mish activation | High accuracy and speed | Advanced applications |
| YOLOv5 | Ultralytics | Pre-trained models, scalable sizes | User-friendly, scalable | Broad industry use |
| YOLOv6 | Optimized for industry | Practical deployment | Efficient and fast | Industrial applications |
| YOLOv7 | New architecture | Latest strategies | State-of-the-art performance | Cutting-edge use cases |
| YOLOv8 | Latest techniques | Modern AI integration | Top performance | Contemporary applications |

**Fig. 2.1**: Comparison of Different YOLO Versions

The YOLO algorithm consists of various versions. Some of the best versions are YOLOv3 and YOLOv4 explained below.

**YOLOv3**

Only Convolution layers are used in YOLOv3[RF16] that's why we call it fully convolutional network (FCN). It includes 75 convolutional layers, including some skip connections, upsampling and down sampling. YOLOv3 not uses any kind of pooling. convolutional layer with a stride of 2 is used to downsample the feature maps. This avoids the loss of low-level features often associated with pooling.

Generally algorithms that use FCN is affected by the input size of the image but this is not true in case of YOLO which is unaffected by the input size of image. However, due to a variety of issues, we may want to stay to a consistent input size in practise. In YOLO if we want to process the imags in batcg wise manner (images in batches may be processed in parallel by the GPU, resulting in speed improvements) then height and width of all the images should be same so, YOLO takes all the images of same size as input which is an issue. Concatenating numerous photos into a huge batch necessitates this.

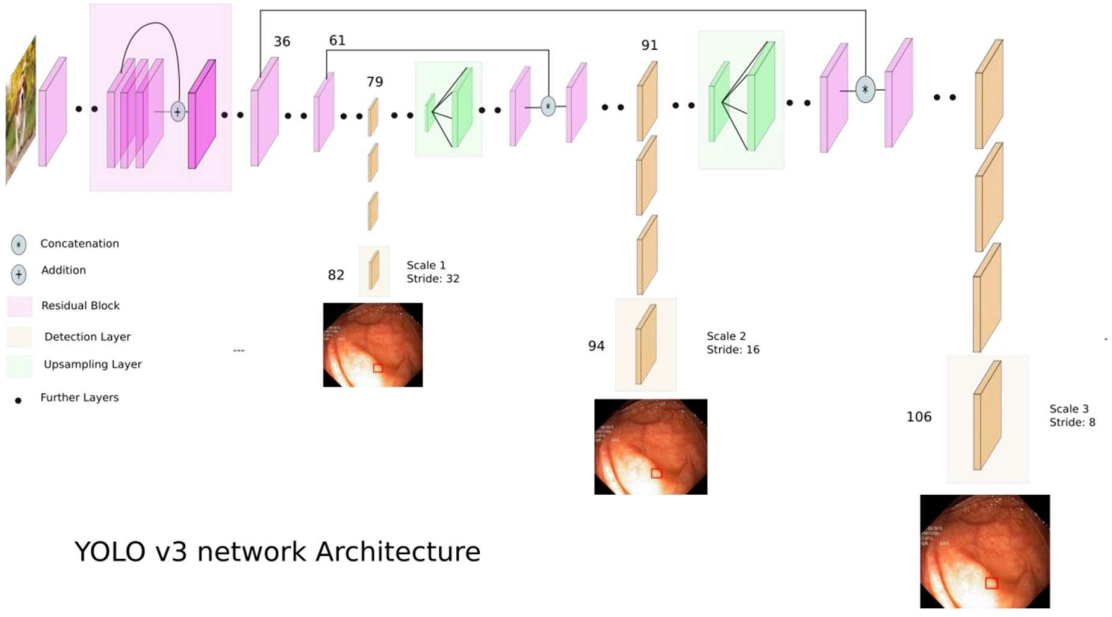


**Fig. 2.2**: YOLOv3 Architecture

**YOLOv3+spp [HW19]**

The difference between yolov3.cfg and yolov3-spp.cfg just this SPP-block:

- In yolov3 configuration file we use downsampling with a stride of 2 in the Convolutional layers.

- In yolov3-spp configuration file we use downsampling with a stride of 2 in the Convolutional layers + gets the best features in Max-Pooling layers.

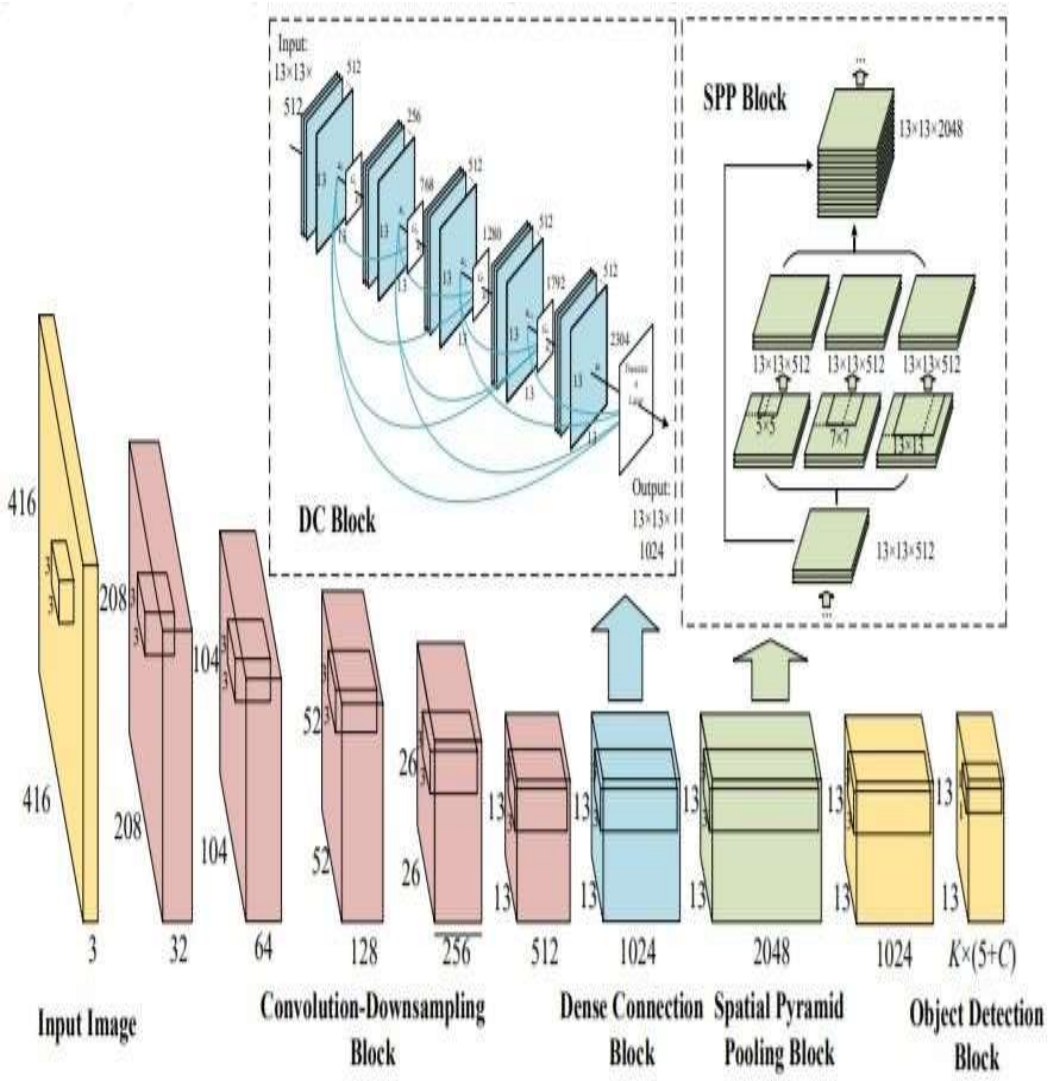


**Fig. 2.3**: YOLOv3+spp Architecture [HW19]

**YOLOv4 [BWL20]**

Our Object detection models is categorized into two parts: one-stage and two-stage models. A one-stage model may recognise objects without the requirement for a pre-processing phase. On the contrary, A two-stage detector employs a preliminary stage in which important locations are identified and classified to determine if an object has been spotted in these areas. A onestage detector has the advantage of being able to produce predictions quickly, allowing for realtime use.

**Why YOLOv4?**

YOLOv4 is an important improvement of YOLOv3, mAP(mean Average Precision) has improved by 10% and the number of FPS(Frame per Second) improved by 12% it happens because of the implementation new architecture. In new architecture the modification is there is the backbone and the neck as well. That makes

YOLOv4 easier to train the network is a single GPU.

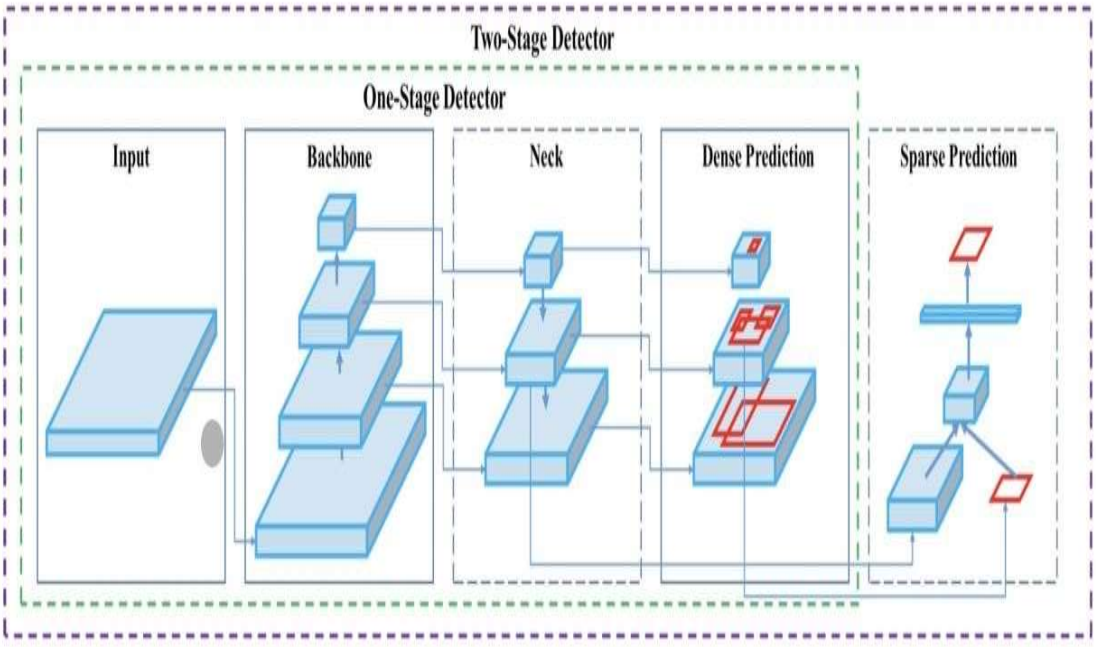


**Fig. 2.4**: YOLOv4 Architecture [BWL20]

**Advantages**

- The main advantage of using YOLO is its speed which is very fast and the frame rate can go upto 45 frames per second.

**Disadvantages**

- When compared to Faster R-CNN, it has a lower recall and a higher localiza- tion error.
- Because each grid can only suggest two bounding boxes, it has difficulty detecting nearby items.
- Struggles to detect small objects.

### 2.2.2 Faster RCNN

Faster R-CNN[RHGS15] is a single-stage, end-to-end trained model. selective search is a very old method and takes very much time time to generate regions so, in order to save time faster RCNN uses the concept of RPN (region proposal network) to generate region proposals. ROI (Region of Interest) Pooling layer basically used to extract feature vectors of fixed length from each region in faster RCNN.

Only a single feature map is required by ROI pooling for all the proposals generated by RPN in a one pass. Faster RCNN requires a fixed size image for object detection which is a problem but ROI pooling also solves this problem. The entire image is fed into a CNN model in order to identify ROI on feature maps.

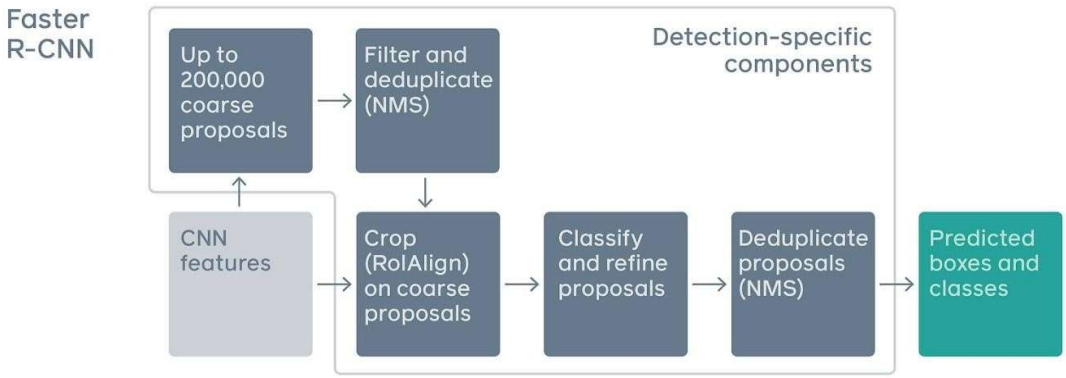


**Fig. 2.5**: Faster R-CNN Architecture

Working of Faster R-CNN:

- Input goes into CNN that generates Feature maps.
- The RPN generates region proposals.
- Using the ROI Pooling layer, a fixed-length feature vector is extracted from each region proposal in the image.
- The Fast R-CNN is then used to classify the extracted feature vectors.
- It finally returns the the class scores of the detected objects along with their boundingboxes.

Faster R-CNN is a widely used object detection framework that builds on the foundations of earlier models such as R-CNN and Fast R-CNN. It introduces a Region Proposal Network (RPN) to generate high-quality region proposals directly from the convolutional feature maps, streamlining the detection process. Here's a concise overview of its architecture and working:

**1. Overall Architecture**

Faster R-CNN consists of three main components:

- Convolutional Backbone: Extracts feature maps from the input image.
- Region Proposal Network (RPN): Proposes candidate object regions.
- RoI Pooling and Detection Network: Refines the proposals and performs object classification and bounding box regression.

**2. Convolutional Backbone**

- Feature Extraction: The input image is passed through a convolutional neural network (CNN), typically using architectures like VGG16, ResNet, or similar. This CNN extracts rich hierarchical features from the image, resulting in a set of convolutional feature maps.
- Shared Backbone: The same feature maps are used by both the RPN and the detection network, enabling shared computation and reducing redundancy.

**3. Region Proposal Network (RPN)**

- Sliding Window: The RPN slides over the feature maps using a small network. At each position, it generates a set of region proposals (anchors) of different scales and aspect ratios.
- Anchor Boxes: Predefined boxes of various sizes and aspect ratios are placed over the feature map to predict object locations. Typically, 9 anchors (3 scales and 3 aspect ratios) are used at each position.
- Proposal Generation: For each anchor, the RPN predicts two outputs:

- Objectness Score: Probability of the anchor being part of an object (foreground) or background.
- Bounding Box Regression: Refinement of the anchor box coordinates to better fit the object.

**4. RoI Pooling**

- Region of Interest (RoI) Pooling: The proposed regions from the RPN are projected onto the feature maps. RoI pooling converts these regions of different sizes into fixed-size feature maps (e.g., 7x7).
- Fixed-Size Representation: This allows the detection network to process each region uniformly, regardless of the original size of the proposals.

**5. Detection Network**

- Classification and Regression: The pooled feature maps are fed into fully connected layers for:
- Object Classification: Predicting the class of the object within each proposal.
- Bounding Box Refinement: Further refining the coordinates of the bounding boxes.

**6. Training Process**

- Multi-Task Loss: Faster R-CNN uses a multi-task loss to jointly train the RPN and the detection network:
- RPN Loss: Combines the classification loss (object vs. background) and the regression loss (anchor adjustment).

- Detection Network Loss: Combines the classification loss (object class) and the regression loss (bounding box refinement).

**7. Inference**

- Region Proposals: During inference, the RPN generates region proposals.
- Non-Maximum Suppression (NMS): Applied to filter out overlapping and redundant proposals, keeping the most likely ones.
- Final Detection: The detection network classifies the objects and refines the bounding boxes for the final predictions.

One disadvantage of Faster R-CNN is that the RPN is trained using a single image to extract all anchors in the mini-batch of size 256. Because all samples from a single image might be linked (i.e. their features are similar), It may take a long time for the network to reach convergence.

One of the main disadvantage of YOLO and Faster R-CNN is that they employ many handcrafted components (Liu et al., 2020), e.g., **non-maximum suppression (NMS)** which is post-processing module, anchor box generation and rule-based training target assignment. So, due to these hand-crafted components they are not fully end-to-end.

Anchor boxes are nothing but a set of bounding boxes with a specific height and breadth or we can say predefined bounding boxes. These boxes are frequently chosen based on the size of the objects in the training datasets to capture the scale and aspect ratio of distinct object classes that must be detected.

Numerous image detections or we can say multiple detections is a problem so NMS a handcrafted method is used to solve this type of problem. In an image nearby cells may detect the same object same as current cell. So, NMS is used to eliminate multiple detections.

These hand-crafted components are very difficult to implement and introduces a lot of complexity.

## 2.3 Dataset

We have used the Kvasir-SEG [JSR+19a] for detection and localization tasks. The image, ground truth data, and their detection are shown in Fig 2.5. (their localised bounding boxes in red). It contains 1000 polyp images, masks and bounding box information taken using a high resolution electromagnetic imaging technology, such as ScopeGuide, Olympus Europe. The segmentation work can be done with the images and their ground facts, whereas the detection task can be done with the bounding box information. The dataset can be downloaded at https://datasets.simula.no/kvasir-seg/.

Images consists of different sized polyps. there are 700 images in which polyps

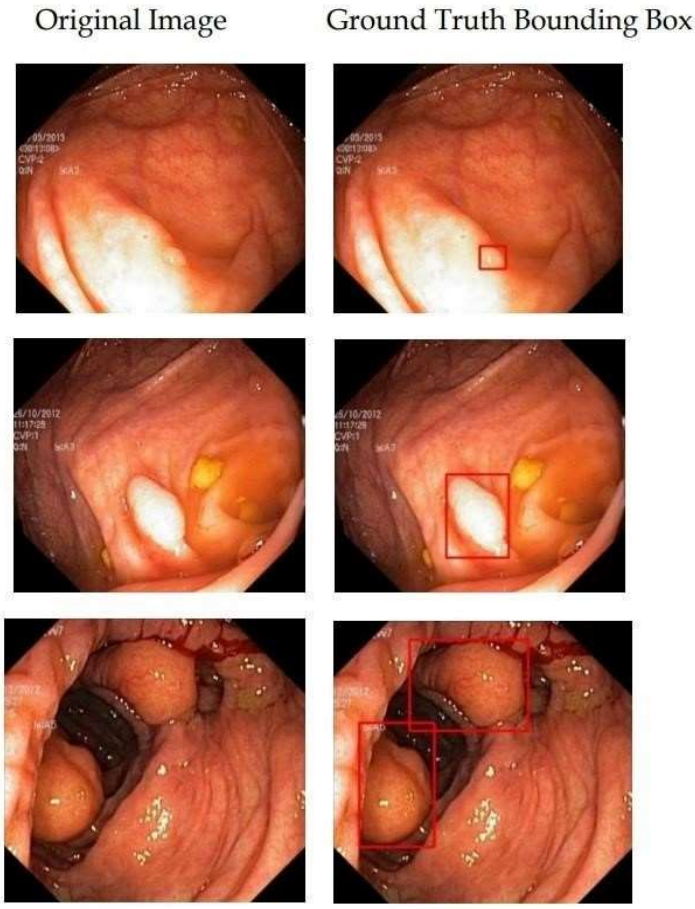


**Fig. 2.6**: Sample images from Kvasir-SEG dataset: Original Image(1st column) and Ground truth bounding boxes (3rd column) for selected samples

are big greater than 160 x 160 pixels, there are 323 polyps that are medium in size that lies somewhere around 64 X 64 pixels and 160 X160 pixels and 48 small sized pixel around 64 x 64 pixels. there are total 1072 polyps along with their bounding box and segmentation mask.

# CHAPTER 3

# PROPOSED APPROACH

In this section, our aim is to propose a model for polyp detection. The model architecture is shown in fig 3.3. In this first a backbone is used to generate feature maps and then these feature maps are flattened so that we can give these feature maps as input to the transformer. At the end the loss function is used transforms the objection detection problem into set-prediction problem and the aim of the loss function is to find the matching with minimum loss.

## 3.1 Detection Transformer

The Detection Transformer, often known as DETR[CMS+20] is a set-based object detector that uses a Transformer on top of a convolutional backbone. It learns a 2D representation of an input image using a traditional CNN backbone. Before feeding data into a transformer encoder, the model flattens it and adds a positional encoding.

DETR (Detection Transformer) is a novel approach to object detection introduced by Facebook AI Research. Unlike traditional object detection models that rely on convolutional neural networks (CNNs) and complex post-processing steps like non-maximum suppression (NMS), DETR leverages a transformer architecture to directly predict the bounding boxes and class labels of objects in an image.

**Key Features of DETR:**

1. Transformer Architecture:

- Attention Mechanism: Uses multi-head self-attention layers to capture long-range dependencies within the image.
- End-to-End Framework: Eliminates the need for hand-crafted components like anchor generation and NMS.

2. Simplified Pipeline:

- Direct Prediction: The model predicts a fixed set of object detections, each represented by a bounding box and a class label, in a single forward pass.
- Positional Encoding: Adds positional information to the image features to help the model distinguish between different parts of the image.

3. Set-Based Los:

- Bipartite Matching Loss: Uses a unique loss function that matches predicted objects with ground truth objects using the Hungarian algorithm, ensuring each object is detected only once.

4. Robustness to Occlusions and Overlapping Objects:

- Global Context Understanding: The attention mechanism allows DETR to understand the global context of the image, improving detection performance, especially in complex scenes with overlapping objects.

**Advantages of DETR**

- Simplicity: The end-to-end nature of DETR simplifies the object detection pipeline, making it easier to train and implement.
- Accuracy: Competitive accuracy compared to traditional CNN-based detectors, especially in detecting overlapping objects.
- Flexibility: Can be easily extended to other tasks such as panoptic segmentation.

**Disadvantages of DETR:**

- Training Time: Requires a longer training period compared to CNN-based detectors due to the complexity of the transformer architecture.
- Computational Resource: Higher computational cost during training and inference because of the extensive use of attention mechanisms.

**Applications of DETR:**

1. Autonomous Driving:

- Obstacle Detection: Detecting and classifying obstacles like pedestrians, vehicles, and traffic signs.
- Scene Understanding: Improved understanding of complex driving environments.

2. Surveillance Systems:

- Security Monitoring: Real-time detection of suspicious activities and objects.
- Crowd Management: Monitoring and managing large crowds in public spaces.

3. Healthcare:

- Medical Imaging: Detecting anomalies and regions of interest in medical scans.
- Surgical Assistance: Assisting surgeons with real-time detection during procedures.

### 3.1.1 Why DETR?

Many modern object detectors like YOLO and Faster R-CNN employ many handcrafted components (Liu et al., 2020), e.g., **non-maximum suppression (NMS)** which is postprocessing module, anchor box generation and rule-based training target assignment. So, due to these hand-crafted components they are not fully end-to-end. These hand-crafted components are very difficult to implement and introduces a lot of complexity that significantly degrade the performance of the model. So, the DETR effectively removes the need of many hand-designed components to improve the performance.

It also uses the concept of transformer on the top of it and transformers has a potential to replace CNN. Originally transformers are meant for NLP (Natural Language processing) but they are performing well in computer vision as well. According to this paper[DBK+20] the Vision Transformer (ViT) is a very powerful tool and has a potential to be used as an alternative to CNN which is currently state-of-the-art in the field of computer vision and that's why it is widely used in image related task. In terms of computational efficiency and accuracy, ViT models exceed the present state-of-the-art (CNN) by nearly four times.

The field of object detection has long been dominated by models based on convolutional neural networks (CNNs), such as Faster R-CNN, SSD, and YOLO. These models have achieved significant success, yet they often rely on complex post-processing steps and hand-crafted design choices like anchor boxes and non-maximum suppression (NMS). DETR (Detection Transformer), introduced by Facebook AI Research, offers a fundamentally different approach that leverages transformer architectures, promising a simpler, end-to-end object detection

pipeline. Here's an in-depth look at why DETR is a groundbreaking advancement in object detection:

**1. Transformer Architecture in Object Detection**

Transformers, originally designed for natural language processing tasks, have revolutionized the way we handle sequential data by effectively capturing long-range dependencies through self-attention mechanisms. DETR adapts this architecture to the task of object detection, using a transformer encoder-decoder structure to process image features and directly predict bounding boxes and class labels.

**Key Advantages:**

- Global Context Understanding: The self-attention mechanism allows DETR to consider the entire image context when making predictions, leading to better performance, especially in scenes with complex relationships between objects.
- Positional Encoding: DETR incorporates positional encodings to retain spatial information, which is crucial for accurate object localization in images.

**2. End-to-End Detection Framework**

Traditional object detection pipelines are often multi-stage, involving separate steps for region proposal, classification, bounding box regression, and post-processing (e.g., NMS). DETR simplifies this by using a single, unified model to handle the entire detection process in an end-to-end manner.

**Key Benefits:**

- Simplified Pipeline: By removing the need for hand-crafted components like anchor boxes and NMS, DETR reduces the complexity of the detection pipeline, making it easier to train and implement.

- Unified Training Objective: The end-to-end approach allows for a more straightforward optimization process, as the entire model can be trained with a single loss function.

**3. Set-Based Prediction and Bipartite Matching Loss**

DETR predicts a fixed set of object detections per image, each represented by a bounding box and a class label. To ensure unique and accurate detections, it uses a set-based loss function, where predictions are matched with ground truth objects using the Hungarian algorithm.

**Key Features:**

- Unique Detections: The bipartite matching ensures that each ground truth object is paired with only one prediction, eliminating duplicate detections.
- Matching Cost: The matching process is based on the overall similarity between predicted and ground truth objects, considering both bounding box coordinates and class probabilities.

**4. Robustness to Occlusions and Overlapping Objects**

One of the challenges in object detection is dealing with occlusions and overlapping objects, where traditional models might struggle to distinguish closely positioned items. DETR's global context awareness and attention mechanism enable it to handle such scenarios more effectively.

**Performance Insights:**

- Occlusion Handling: By considering the entire image context, DETR can better infer the presence of occluded objects and accurately predict their locations.
- Overlapping Objects: The self-attention mechanism helps DETR to distinguish between overlapping objects, leading to fewer missed or misclassified detections.

**5. Flexibility and Extensibility**

DETR's transformer-based architecture and end-to-end training make it highly flexible and extensible to various computer vision tasks beyond object detection, such as panoptic segmentation and keypoint detection.

**Applications:**

- Panoptic Segmentation: Combining object detection and semantic segmentation into a unified framework.
- Keypoint Detection: Extending the model to predict keypoints for tasks like pose estimation.

**6. Training Dynamics and Computational Requirements**

While DETR offers several advantages, it also comes with certain challenges. One notable issue is the longer training time compared to traditional CNN-based detectors, primarily due to the extensive computation required by the transformer architecture.

**Considerations:**

- Training Time: DETR requires more epochs to converge, necessitating careful consideration of computational resources.
- Resource Intensity: The self-attention mechanism, while powerful, is computationally expensive, especially for high-resolution images.

**7. Comparative Performance**

Empirical evaluations have shown that DETR achieves competitive performance with state-of-the-art CNN-based detectors on benchmarks like COCO. It excels particularly in scenarios involving complex object relationships and cluttered scenes.

**Evaluation Metrics:**

- Precision and Recall: DETR demonstrates high precision and recall, particularly for large objects and scenes with multiple overlapping items.
- Latency: While inference speed is generally slower due to the transformer architecture, optimizations and model pruning techniques can help mitigate this.

Architecture of DETR is also very simple. So, due to these reasons we have used Detection Transformer for our study.

Transformer uses self-attention mechanism which basically calculates attention of all other inputs with respect to one input.

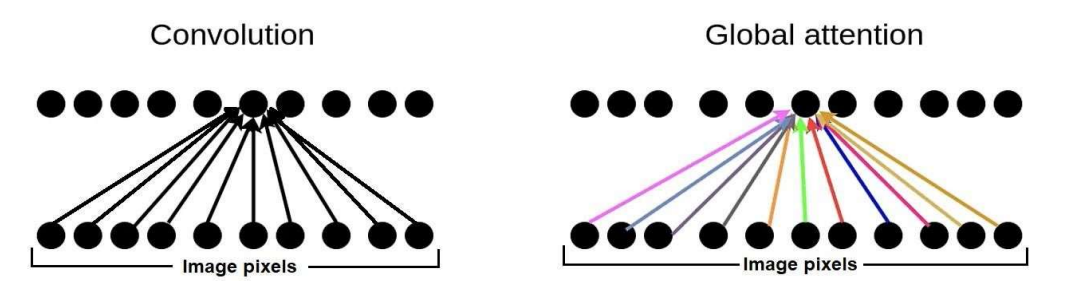


**Fig. 3.1**: Difference between convolution and attention

So, first let's understand the difference between convolution and attention. In convolution we basically applying different filters over the entire image and the same filter is applied to every pixel in the image that means in order to calculate next output we give equal focus to each and every pixel. But in case of self-attention mechanism we focus only on the important regions or we can say give more attention on those pixels that are relevant to it shown in fig 3.1.

### 3.1.2 Architecture of Detection Transformer

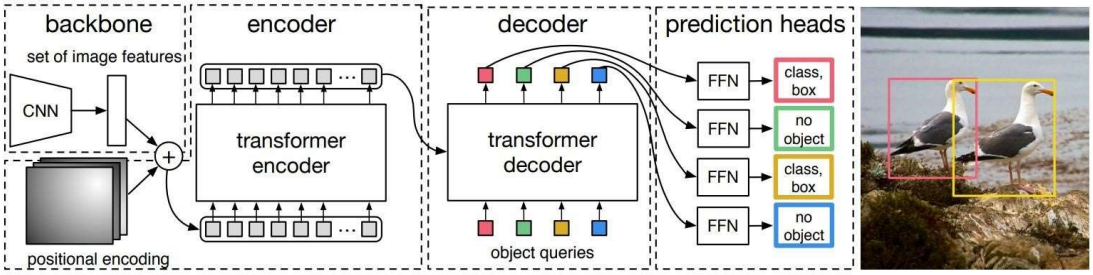


**Fig. 3.2**: DETR Architecture [CMS+20]

The architecture of DETR is very simple. It basically contains three main components:

- CNN backbone
- Encoder-Decoder transformer
- simple feed forward network (FFN)

**Backbone.** We adopt ResNet101 [HZRS15] as a feature extractor backbone. It takes as input a image and outputs a high-level feature map. Generated feature maps are concatenated with positional encoding (it basically describes the location or position of each and every pixel with unique representation) and then we flatten all the pixels to make input sequence that is acceptable by the transformer encoder. **Transformer Encoder.** The flatten input sequence generated by backbone is then given to encoder. In the encoder we have N number of layers in each layer we perform multi-head attention (just like convolution filters here we have several heads which takes Queries, key and value as input) and Feed forward Network (FFN). The output of encoder is the combination of keys and values which is then given input to the decoder.

**Transformer Decoder.** It takes object queries and key-value pair generated by encoder. It makes use of standard architecture of the transformer which contains N layers. It includes

masked multi head attention, which ensures that predictions for present position can be based solely on known outputs at positions lower than this. It generates Q outputs which then passed through Q FFN's (Feed forward Network) and each FFN generates an output which contains class of the object and the bounding box coordinates in the form (xmin, ymin, xmax, ymax). We can find the detailed architecture of DETR transformer which is slightly different as compared to the original transformer in Fig 3.3.

### 3.1.3 Loss Function

Here the loss produces an optimal bipartite matching that has minimum loss between predicted and ground truth objects, and then optimize the object-specific (bounding box) losses. Loss is shown below:

$$\hat{\sigma} = \underset{\sigma \in S_N}{\operatorname{argmin}} \sum_{i}^{N} L_{match}(y_i, \hat{y}_{\sigma(i)})$$

where $L_{match}(y_i, \hat{y}_{\sigma(i)})$ is the pairwise matching cost between ground truth box and prediction and $\hat{\sigma}$ is the optimal assignment. To find the optimal assignment hungarian algorithm is used.

$$L_{match}(y_i, \hat{y}_{\sigma(i)}) = -1_{\{c_i \neq \phi\}} \hat{p}_{\sigma(i)}(c_i) + 1_{\{c_i \neq \phi\}} L_{box}(b_i, \hat{b}_{\sigma(i)})$$

where $-1_{\{c_i \neq \phi\}} \hat{p}_{\sigma(i)}(c_i)$ is the class loss and $1_{\{c_i \neq \phi\}} L_{box}(b_i, \hat{b}_{\sigma(i)})$ is the bounding box loss.

**Bounding Box Loss.** Bounding box is calculated by adding $l_1$ loss and the Generalized Intersection over union.

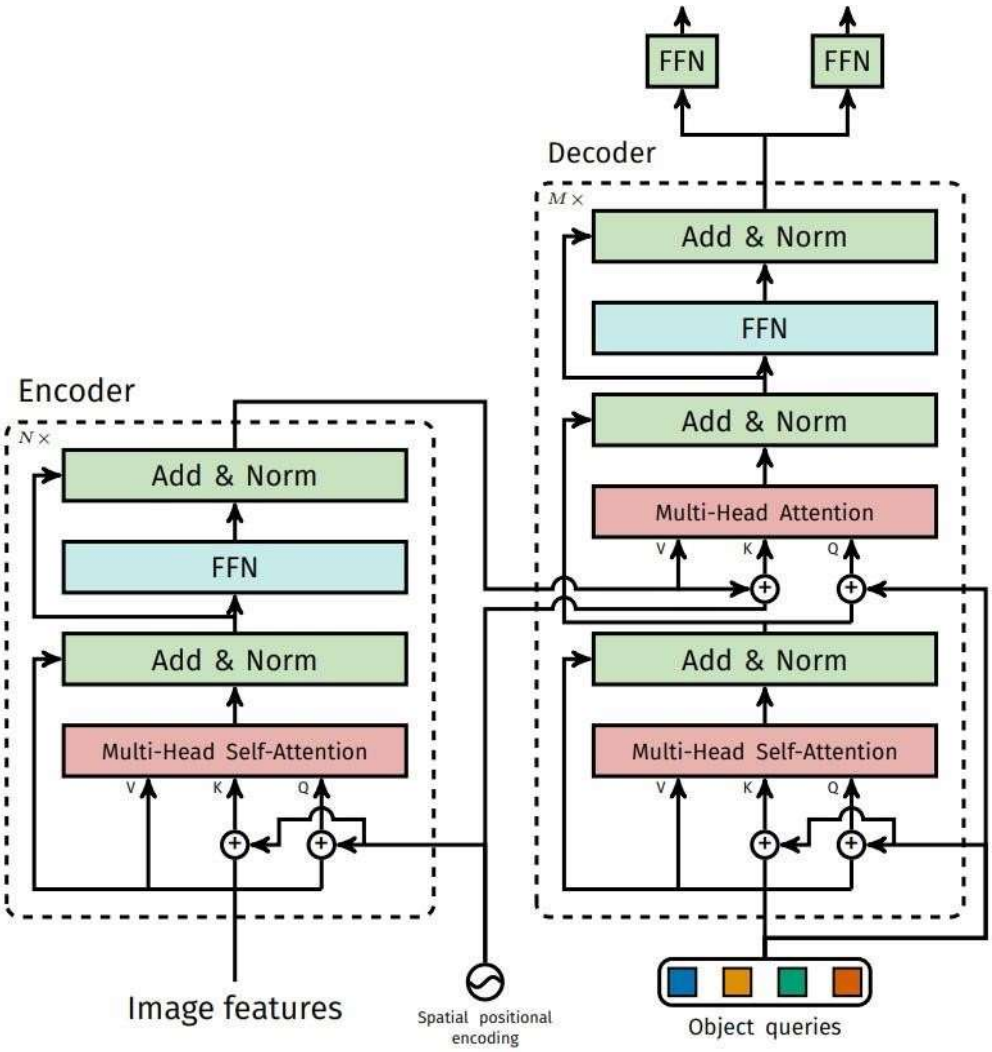


**Fig. 3.3**: DETR Transformer Architecture [CMS+20]

$$L_{box}(b_i, \hat{b}_{\sigma(i)}) = \lambda_{iou} L_{box}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{L1} ||b_i - \hat{b}_{\sigma(i)}||$$

This is all about DETR but there are two problems in DETR:

- It takes much longer training time to converge as compare to existing object detectors.

- It is not good in detecting small objects.

So, in order to improve these problems Chinese University of Hong Kong proposed a model known as Deformable DETR which is an improvement of DETR and very similar to DETR with minor modifications.

### 3.2 Deformable DETR

In Deformable DETR [ZSL+20] we are using multi-scale multi head attention instead of using single scale attention which is responsible for detecting small objects.

In DETR for calculation of current pixel value we focus on all the pixels which is very costly but in Deformable DETR we only focus on small set of locations or pixels on different scales to calculate current value which reduces the training time. We can see the new architecture of transformer in Fig 3.4.

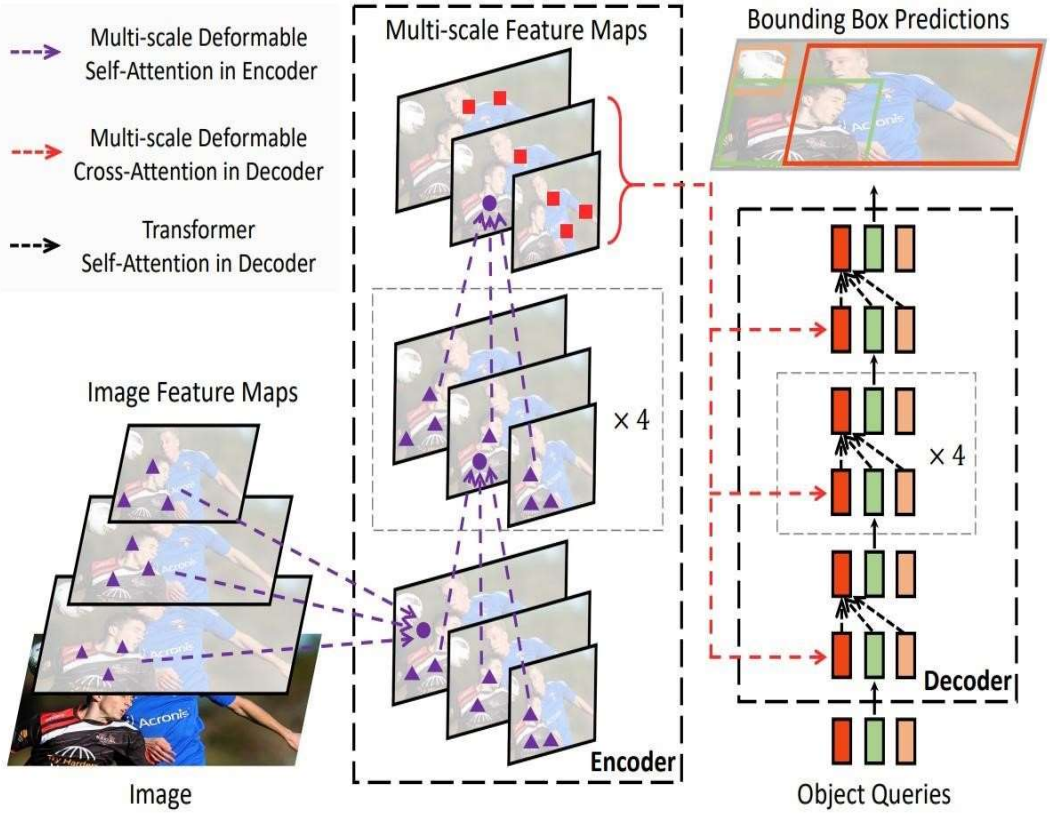


**Fig. 3.4**: Deformable DETR Transformer Architecture [ZSL+20] Deformable DETR has some additional improvements as well:

**Iterative Bounding Box Refinement.** In this we pool the features from previous object location predictions iteratively.

**Two-Stage Deformable DETR.** In this they basically use a variant of Deformable DETR that generates region proposal and then fed these region proposals into decoder as object queries for further refinement.

**Difference Between DETR and Deformable DETR**

DETR (Detection Transformer) and Deformable DETR are both innovative approaches to object detection that utilize transformer architectures. While DETR introduced a novel end-to-end framework for object detection, Deformable DETR was developed to address some of DETR's limitations, particularly regarding efficiency and performance on small objects. Here's a detailed comparison of the two models:

**1. Architecture and Design**

**DETR:**

- Transformer-Based Architecture: Uses a standard transformer encoder-decoder setup.
- Global Self-Attention: Applies self-attention mechanisms to the entire image, capturing global context but at a high computational cost.
- Positional Encoding: Incorporates positional encodings to retain spatial information.
- Fixed Number of Predictions: Predicts a fixed set of bounding boxes and class labels in a single forward pass.

**Deformable DETR:**

- Deformable Attention Mechanism: Introduces deformable attention modules that focus on a sparse set of key sampling points around a reference point, significantly reducing computational complexity.
- Efficiency: By limiting attention to local regions, Deformable DETR is more efficient and scalable.
- Enhanced Positional Encoding: Uses multi-scale deformable attention to better handle objects at different scales.
- Improved Handling of Small Objects: The deformable attention mechanism improves detection accuracy for small and densely packed objects.

**2. Performance and Speed**

**DETR:**

- Training Time: Requires longer training times to converge compared to traditional CNN-based detectors due to the complexity of the global attention mechanism.
- Inference Speed: Relatively slower during inference because of the computational intensity of global self-attention.

**Deformable DETR:**

- Training Efficiency: Converges faster than DETR due to the efficient attention mechanism.
- Inference Speed: Faster inference times as the deformable attention mechanism reduces computational overhead.
- Performance on Small Objects: Shows improved performance on small and medium objects compared to the original DETR.

**3. Loss Function and Optimization**

**DETR:**

- Bipartite Matching Loss: Uses a set-based loss function with the Hungarian algorithm to match predictions with ground truth objects uniquely.
- Components: The loss function includes classification loss and bounding box regression loss.

**Deformable DETR:**

- Adapted Loss Function: Retains the bipartite matching loss but optimizes it for the deformable attention mechanism.
- Scalability: Better scalability in training large-scale datasets due to reduced computational demands.

**4. Handling Different Scales and Aspect Ratios**

**DETR:**

- Challenges with Scale Variability: Struggles with detecting objects of varying scales, especially small objects, due to its reliance on global attention.
- Aspect Ratio Sensitivity: Can have difficulty with objects of different aspect ratios because of the fixed grid approach.

**Deformable DETR:**

- Multi-Scale Features: Incorporates multi-scale deformable attention to handle objects at various scales more effectively.
- Adaptive Receptive Fields: The deformable attention mechanism dynamically adjusts to the size and shape of objects, improving accuracy across different aspect ratios.

**5. Applications and Use Cases**

**DETR:**

- General Object Detection: Suitable for a wide range of object detection tasks but particularly excels in scenarios where objects are large and not densely packed.
- Semantic Understanding: Beneficial in applications requiring a comprehensive understanding of the scene, such as panoptic segmentation.

**Deformable DETR:**

- High-Resolution Imagery: More suited for applications involving high-resolution images and scenarios with small, dense objects.
- Real-Time Detection: Better suited for real-time object detection tasks due to its faster inference times.

# CHAPTER 4

# EXPERIMENTS AND RESULTS

### 4.1 Experimental datasets

The framework used in the proposed work is Pytorch. Tesla P100-PCIE GPU by Google Colab is used for for training the network. Kvasir-SEG dataset contains 1000 images which is splitted into two parts one is for training and other is for testing. training data consists of 880 images where as the testing data consists of 120 images.

## 4.2 Evaluation Metric

Standard computer vision metrics are used to evaluate polyp detection and localization methods on the Kvasir-SEG dataset.

**IoU (intersection over Union).** It's a metric for determining how accurate an object detector is on a specific dataset.

**Recall.** It also called sensitivity. Out of the total actual positive values how many values did we correctly predicted positively or we can say what portion of the true positive are correctly classified.

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

**Precision.** It is also known as True positive prediction value. Out of the total predicted positive results how many results were actually positive or we can say what portion of my positive predictions are correct.

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

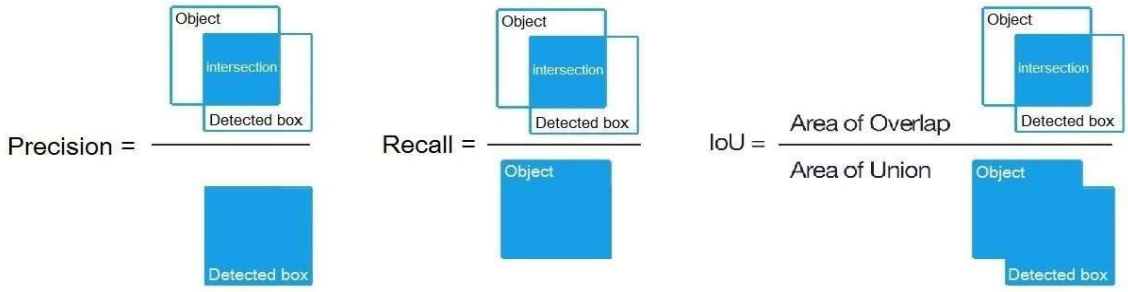


**Fig. 4.1**: Precision, Recall and IoU (intersect over union)

**AP** (Average Precision). It is basically the area under the precision-recall curve at certain IoU threshold. **mAP** (mean Average Precision). It is the average of AP.

### 4.3 Experimental Settings

The experiments have been performed on PyTorch. We trained Deformable DETR with AdamW optimizer setting the initial transformer's learning rate to $1e-4$, the backbone's to $1e{-}5$, and weight decay to $1e{-}4$ after 20 epochs on the total number of epochs of 50.

### 4.4 Encoder-Decoder Layers

This is the first experiment that we have performed to find the optimal number of encoder and decode layers. So, by performing several experiments we found that 6 encoder layers and 6 decoder layers performed best as shown in Fig. 4.2.

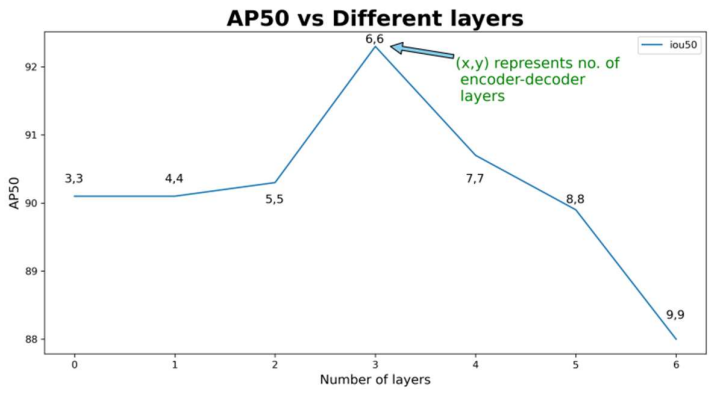


**Fig. 4.2**: Graph between AP at IoU threshold 0.5 and different number of layers

As we can see in the above graph when the encoder-decoder layers are 3,3 the AP50 is 90.1% and when we are increasing the layers till 6,6 AP50 increases (i.e. 92.3% which is optimal) but after that it decreases. So, this proves that the optimal number of layers for encoder and decoder is 6,6.

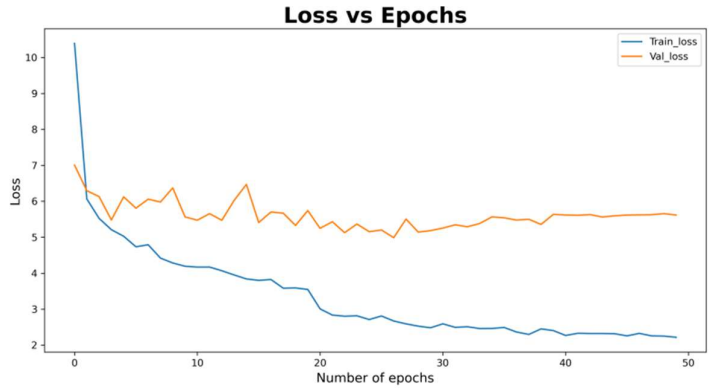


**Fig. 4.3**: Training and Validation loss curve of Deformable DETR

### 4.5 Comparison of Deformable DETR with YOLOv4 Results

YOLOv4 is one of the best state-of-the-art methods. In this we have used Darknet53, CSP as a backbone and we trained YOLOv4 wih SGD optimizer at a Learning rate of $1e{-}4$, batch size is 64 and Loss function is CIoU, CE (CIoU stands for Complete Intersection over union and

CE stands for Binary cross entropy for the objectness and classification scores)

*Note: **green box represents predicted box by the model** and the **Red box represents ground truth box (i.e. original one)**.

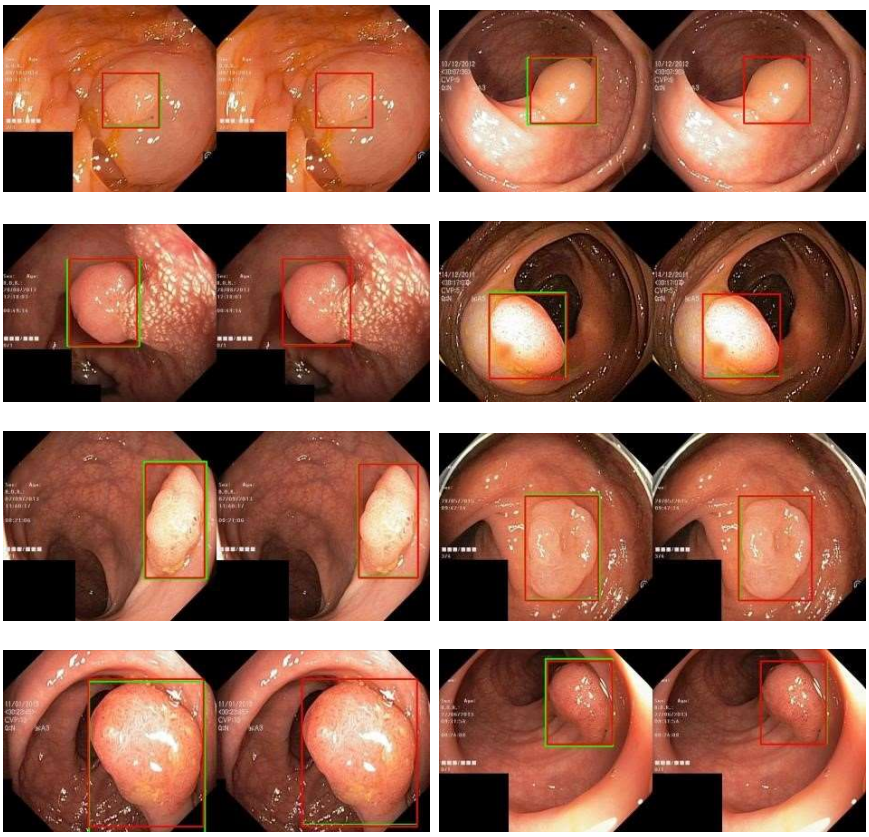

**Fig. 4.4**: Results that are good in both YOLOv4 and Deformable DETR

Figure 4.4 contains 8 images and every image consists of two sub image in which first image is predicted by YOLOv4 and second image is predicted Deformable DETR with GIoU loss.

These results are very good in both the detectors but the Deformable DETR shows more accurate box as seen in the Fig 4.4.

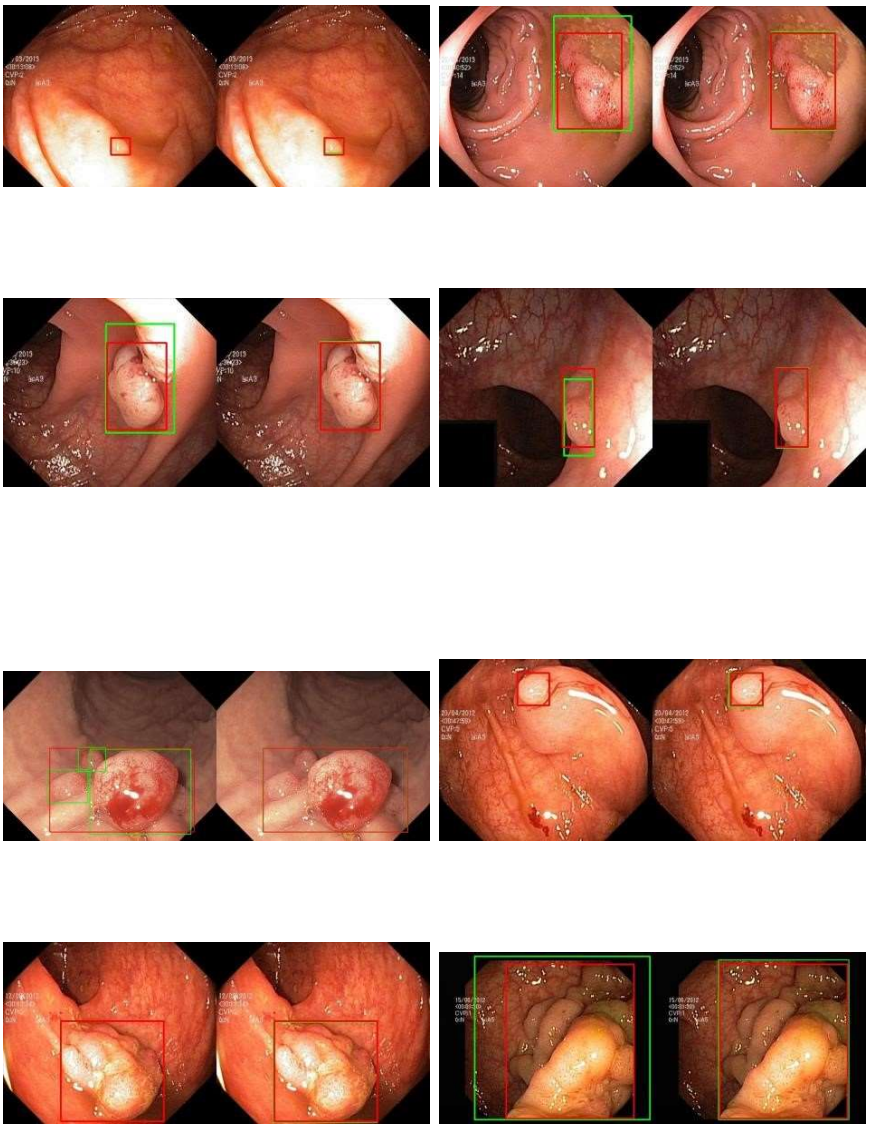

**Fig. 4.5**: Results that are not good in YOLOv4 and but good in Deformable DETR

Figure 4.5 contains 8 images and every image consists of two sub image in which first image is predicted by YOLOv4 and second image is predicted Deformable DETR with GIoU loss.

In Fig 4.5 we have observed that polyps with sharp edge or bigger size are easily detected with YOLOv4 but in case of small objects YOLOv4 is not working but Deformable DETR works better in both the cases. There is one problem with Deformable DETR that it is not good in case when the bounding box have different aspect ratio and when the edges are not clear or sharp.

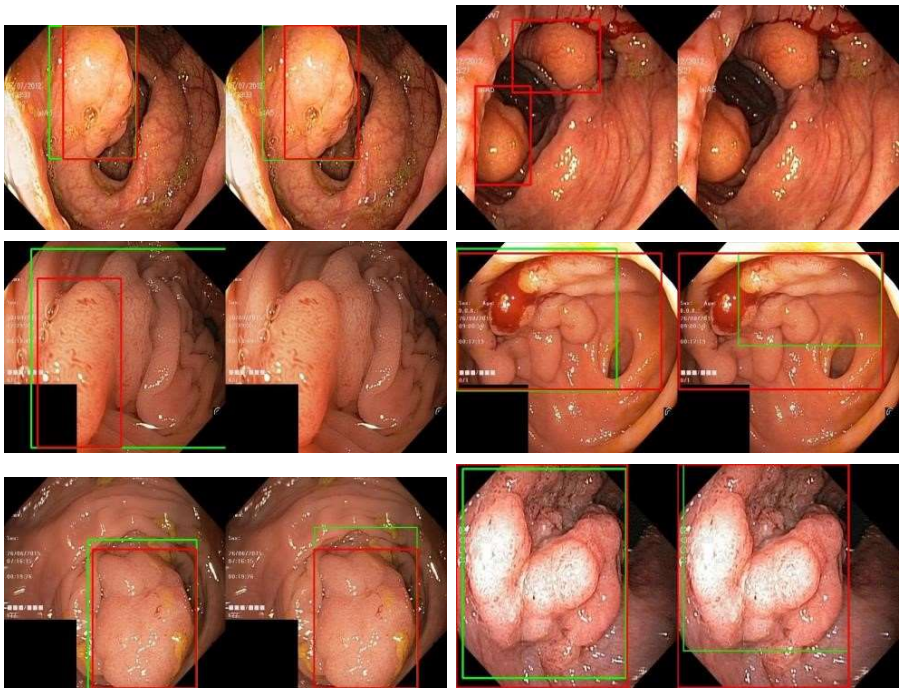

**Fig. 4.6**: Results that are bad in both YOLOv4 and Deformable DETR

Figure 4.6 contains 6 images and every image consists of two sub image in which first image is predicted by YOLOv4 and second image is predicted Deformable DETR with GIoU loss.

In Figure 4.6 images having multiple polyps or having polyp like structure around the polyp in such case Deformable DETR and YOLOv4 both are not detected properly. They either detect half polyp or not even detecting anything in the image.

### 4.6 Loss Function Improvement

While considering previous problem, when we were studying about the loss function of deformable DETR, we came to know that the bounding box loss function that deformable DETR is using is not good for the boxes with different aspect ratio and that is the reason we are not getting good results.

First, let's understand about the Bounding box loss functions [ZWL+19]. Basically there are 4 types of loss functions:

- IoU (Intersection Over Union)
- GIoU (Generalized IoU)
- DIoU (Distance IoU)
- CIoU (Complete IoU)

**IoU (Intersection Over Union).**

It is very basic and simple loss that only works when the predicted bounding boxes overlap with the ground truth box. For non-overlapping cases IoU loss would not provide any moving gradient. The convergence speed is very slow in case of IoU loss.

$$L_{IoU} = \frac{|B \bigcap B^{gt}|}{|B \bigcup B^{gt}|}$$

It fails when predicted and ground truth boxes do not overlap.

**GIoU (Generalized IoU).**

This loss basically tries to maximizes the overlaped area between the predicted box and the the ground truth and bounding box. For non-overlapping cases, it moves the predicted bounding box slowly towards the target bounding box to overlap with the target box.

$$L_{GIoU} = 1 - IoU + \frac{|C - B \bigcup B^{gt}|}{|C|}$$

where C is the smallest box covering both predicted and ground truth bounding boxes, which act like a penalty term moving the predicted box closer to the target ground truth box.

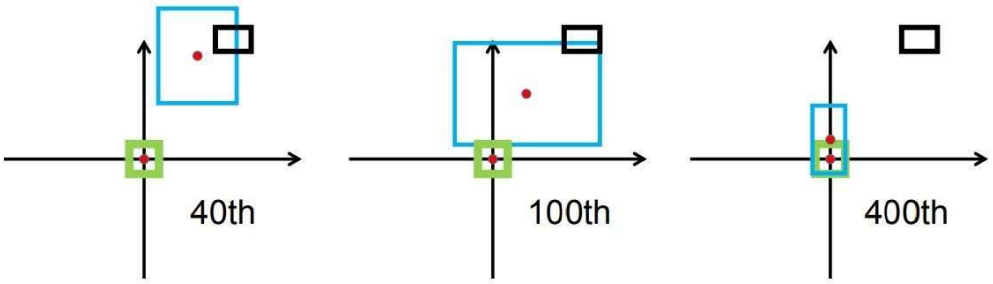


**Fig. 4.7**: GIoU loss: Blue is the predicted bounding box using GIoU loss[ZWL+19]

Initially GIoU loss increases the size of predicted bounding box and moves towards the ground truth box slowly over several iterations to overlap the predicted box to the ground truth box especially when the bounding boxes have a horizontal and vertical orientation. GIoU loss gives better precision than the MSE(mean squared error) loss and IoU loss. For nonoverlapping cases GIoU loss solves the problem of vanishing gradients but the convergence speed is very slow and regression loss is also inaccurate, especially for the boxes with extreme aspect ratios.

**DIoU (Distance IoU).**

The normalized distance between the center point of the predicted boxes and ground truth box is known as DIoU. we can achieve fast convergence and accurate regression wuth the help of this Distance loss.

$$L_{DIoU} = 1 - IoU + \frac{d^2}{c^2}$$

where d represents the Euclidian distance between the center points both the boxes(i.e. ground truth and predicted one), and c is the diagonal length of the box covering both the boxes ground truth and the predicted one as shown in Fig 4.8.

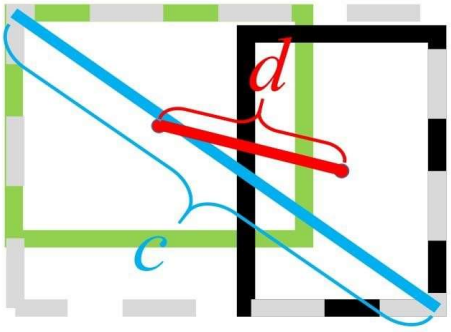


**Fig. 4.8**: d and c between predicted and ground truth box[ZWL+19]

In case for non-overlapping cases DIoU loss also provides the moving directions for predicted bounding boxes .

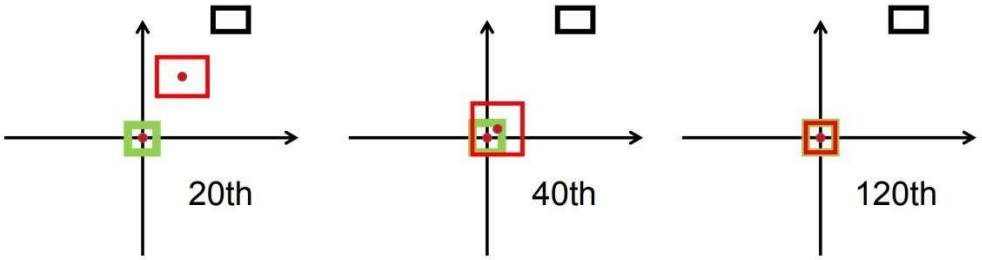


**Fig. 4.9**: DIoU loss: Red is the predicted bounding box using DIoU loss[ZWL+19]

As compared to GIoU loss DIoU loss directly minimizes the distance between predicted and ground truth-bound boxes and also converges much faster even when the ground truth boxes have horizontal and vertical orientations.

**CIoU (Complete IoU).**

It is the complete intersection over union that uses three factors.

- First factor is overlap area between ground truth and predicted bounding box which is also known as IoU loss denoted by $S(B,B^{gt})$.
- Second Factor is DIoU loss that is the central point between ground truth and predicted bounding box denoted by $D(B,B^{gt})$.
- Third factor is the aspect ratio of the predicted box and the ground truth box denoted by $V(B,B^{gt})$.

$$L_{CIoU} = S(B,B^{gt}) + D(B,B^{gt}) + V(B,B^{gt}) \quad (1)$$

For non-overlapping cases CIoU loss also moves the predicted bounding box towards the ground truth bounding box. Convergence rate of CIoU loss is much faster than GIoU loss. It makes regression very fast with extreme aspect ratios. CIoU loss is applied in Faster RCNN, SSD YOLOv3 and YOLOv4.

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v)}$$

$$S(B,B^{gt}) = 1 - IoU \quad (2)$$

$$D(B, B^{gt}) = \frac{d^2}{c^2} \quad (3)$$

$$V(B,B^{gt}) = \alpha \times v \quad (4)$$

in equation 3, v measures the consistency of aspect ratio and $\alpha$ is the trade-off parameter, (5)

(6)

by putting equation 2, 3, 4, 5 and 6 in equation 1 we get the final formulae for

$L_{CIoU}$,

$$L_{CIoU} = 1 - IoU + \frac{d^2}{c^2} + (\frac{v}{(1 - IoU) + v}) \times (\frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2) \quad (7)$$

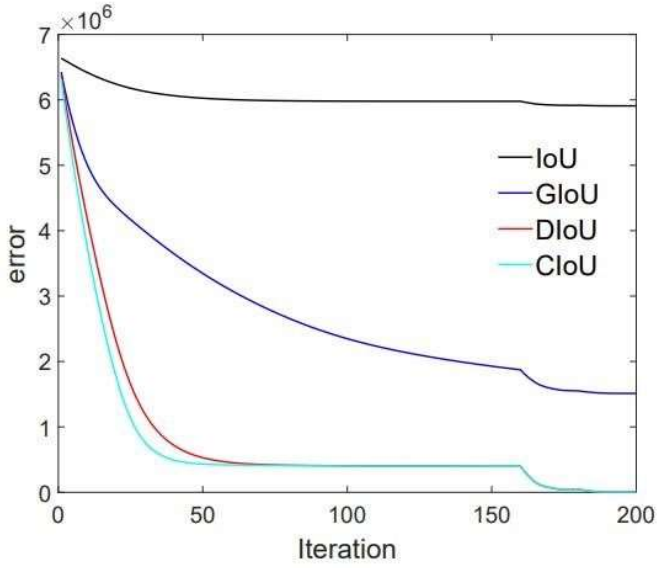


**Fig. 4.10**: regression error sum curves of different loss functions for different iterations[ZWL+19].

Now, Deformable DETR using GIoU as Bounding box loss has been discussed earlier. In here we found one scope of improvement by replacing GIoU loss with CIoU loss. After replacing this we have got better accuracy as well as better convergence rate.

Deformable DETR without CIoU loss has achieved 88.1% mAP in 26 epochs but with CIoU loss we got 88.9% mAP(i.e. increment of 0.8%) in just 8 epochs. So, we can say that by changing loss function, along with better mean average precision we got better convergence rate. Result comparison is shown in Fig 4.11. In these results one image contains three images in which first image is predicted by YOLOv4, second image is predicted by Deformable DETR with GIoU loss and third image is predicted by Deformable DETR with CIoU loss function.

Figure 4.11 contains 8 images and every image consists of three sub image in which first image is predicted by YOLOv4, second image is predicted Deformable

DETR with GIoU loss and third image is predicted by Deformable DETR with

CIoU loss

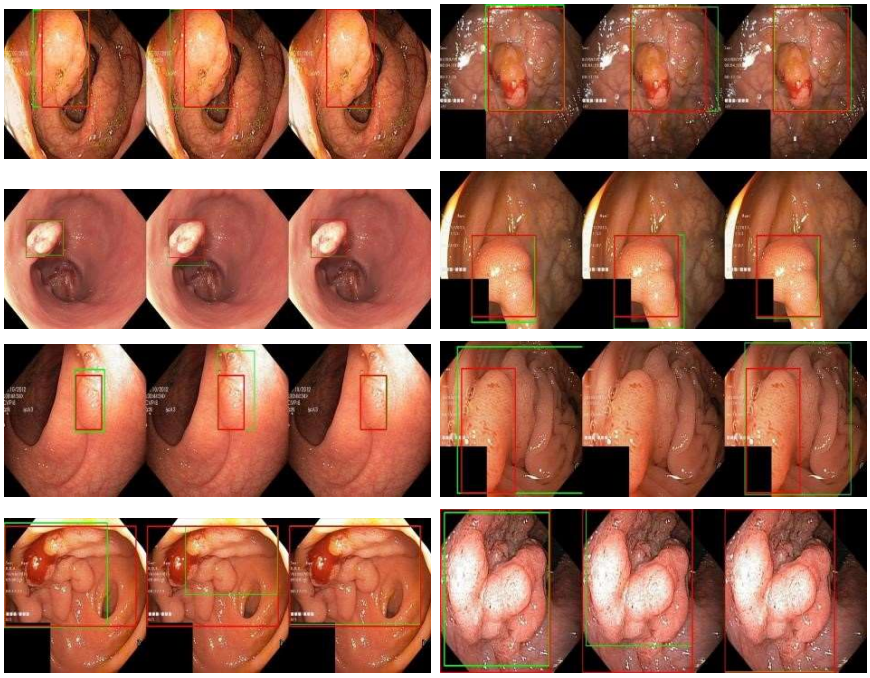

**Fig. 4.11**: Results that are bad in both YOLOv4 and Deformable DETR with

GIoU loss but improved in Deformable DETR with CIoU loss

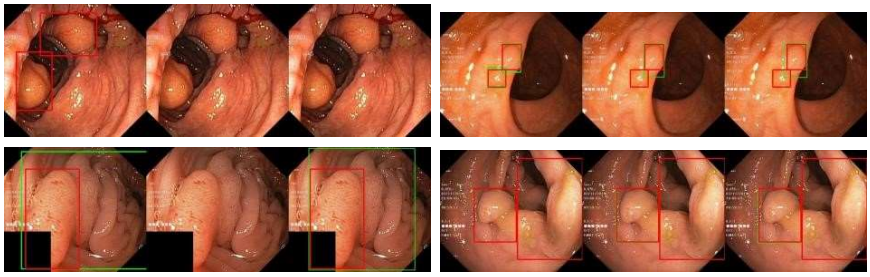

**Fig. 4.12**: Results that are bad even in Deformable DETR with CIoU loss

Figure 4.12 contains 4 images and every image consists of three sub image in which first image is predicted by YOLOv4, second image is predicted Deformable

DETR with GIoU loss and third image is predicted by Deformable DETR with CIoU loss

From Fig 4.12 we can say the Deformable DETR is not good when there are multiple objects in the image.

## 4.7 Image Augmentation

While reading about the transformers we found one drawback. Transformer requires large amount of data to train (around 5k to 10k images atleast) but we provided only have 880 images to train. So, in order to overcome this problem we have adopted some augmentation techniques like Rotation, flipping, shearing, scaling and blurring.

In augmentation, first we have applied three rotations ($90^0$,$180^0$ and $270^0$), second we have applied three type of flipping (horizontal, vertical and horizontalvertical combined), third we have applied two type of shearing on x and y axis, fourth we have applied Zoom in 20% and Zoom out (10%, 30% and 50%) and at the end we have applied blurring. These augmentations are applied on all 880 images and we have got total 12320 images for training.

Figure 4.13 contains 5 images and every image consists of four sub image in which first image is predicted by YOLOv4, second image is predicted Deformable DETR with GIoU loss, third image is predicted by Deformable DETR with CIoU loss and fourth image is predicted by Deformable DETR with CIoU loss with augmentation

## 4.8 Final result comparison

you can see final performance of all three variants of Deformable DETR and compariosn with state-of-the-art methods in Table 4.1.

In table 4.1 MSE is mean squared error, CE is binary cross entropy, CSP is Cross-Stage Partial Connection, GIoU is generalized Intersection Over Union and

CIoU is Complete Intersection Over Union

## 4.9 Test Time Augmentation

We have introduced one new testing strategy to enhance the model performance. we can see the results in Fig 4.14 and Fig 4.15. In this, we have applied the same

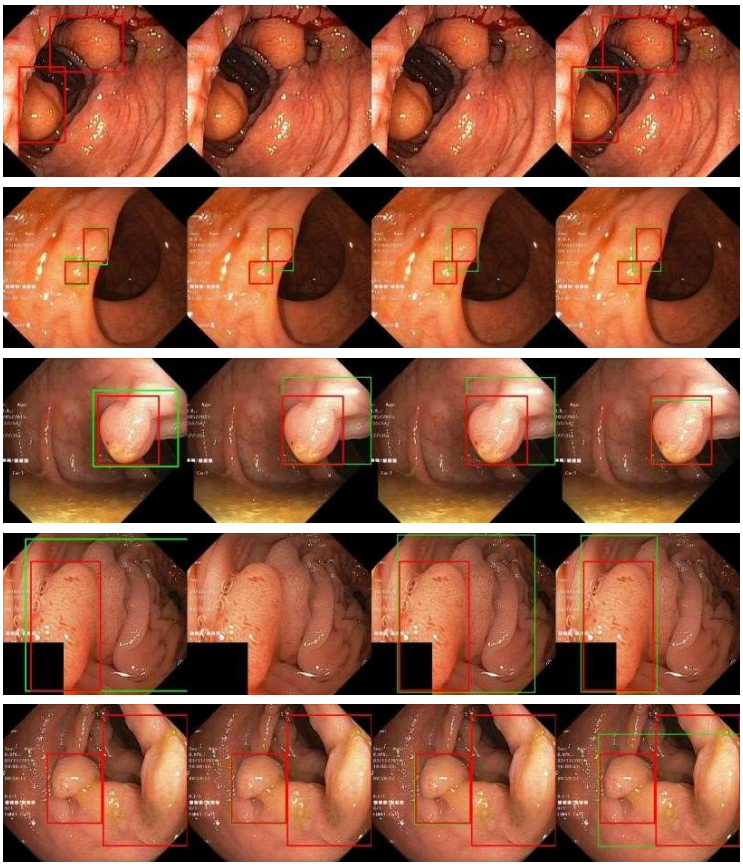

**Fig. 4.13**: Results that are not good earlier but after augmentation they becomes recognisable

augmentation strategies to the test set as well. In test set we have total 120 images. After applying the same augmentation strategies we got total 1680. The detailed results are shown below.

In table 4.2 MSE is mean squared error, CE is binary cross entropy, CSP is

Cross-Stage Partial Connection, GIoU is generalized Intersection Over Union and CIoU is Complete Intersection Over Union

In table 4.2 Deformable DETR with CIoU loss and augmentation gives the better mean average precision. But we have observed one thing that in case of Deformable DETR with CIoU loss and without augmentation, the total number of predictions are more (i.e. 1657 out of 1680) as compared to with augmentation

| Method | Loss Function | Backbone | mAP | AP50 | AP75 |
|---|---|---|---|---|---|
| Faster R-CNN | $L1^{smooth}$, log-loss | ResNet50 | 0.7039 | 0.8418 | 0.566 |
| RetinaNet | $L1^{smooth}$, focal loss | ResNet50 | 0.8031 | 0.9095 | 0.6967 |
| RetinaNet | $L1^{smooth}$, focal loss | ResNet101 | 0.81135 | 0.9095 | 0.7132 |
| YOLOv3 + spp | MSE, CE | Darknet53 | 0.8059 | 0.8532 | 0.7586 |
| YOLOv4 | CIoU, CE | Darknet53, CSP | 0.8239 | 0.8929 | 0.7549 |
| Deformable DETR | $L1^{smooth}$, GIoU, CE | ResNet101 | 0.8810 | **0.9230** | 0.8390 |
| Deformable DETR | $L1^{smooth}$, **CIoU**, CE | ResNet101 | **0.8890** | 0.9150 | **0.8630** |
| Deformable DETR(Aug) | $L1^{smooth}$, **CIoU**, CE | ResNet101 | 0.8660 | 0.9050 | 0.8270 |

**Table 4.1**: Result on the polyp detection task on the Kvasir-SEG dataset and best score is highlighted in last 3 columns.

| Method | Loss Function | Backbone | mAP | AP50 | AP75 |
|---|---|---|---|---|---|
| Deformable DETR | $L1^{smooth}$, GIoU, CE | ResNet101 | 0.7410 | 0.7880 | 0.6940 |
| Deformable DETR | $L1^{smooth}$, **CIoU**, CE | ResNet101 | 0.7370 | 0.7900 | 0.6840 |
| Deformable DETR(Aug) | $L1^{smooth}$, **CIoU**, CE | ResNet101 | **0.7730** | **0.8290** | **0.7170** |

**Table 4.2**: Result on the polyp detection task on the Augmented Kvasir-SEG dataset and best score is highlighted in last 3 columns

(i.e. 1628 out of 1680) and In case of Deformable DETR with GIoU loss number of predictions are 1637 out of 1680.

In Fig 4.14 and 4.15 first row represents Rotation($0^0$,$90^0$,$180^0$,$270^0$), second row represents different flipping, third row represents shearing on x and y axis respectively, fourth row represents Zoom in and Zoom out and last row is for blurred image.

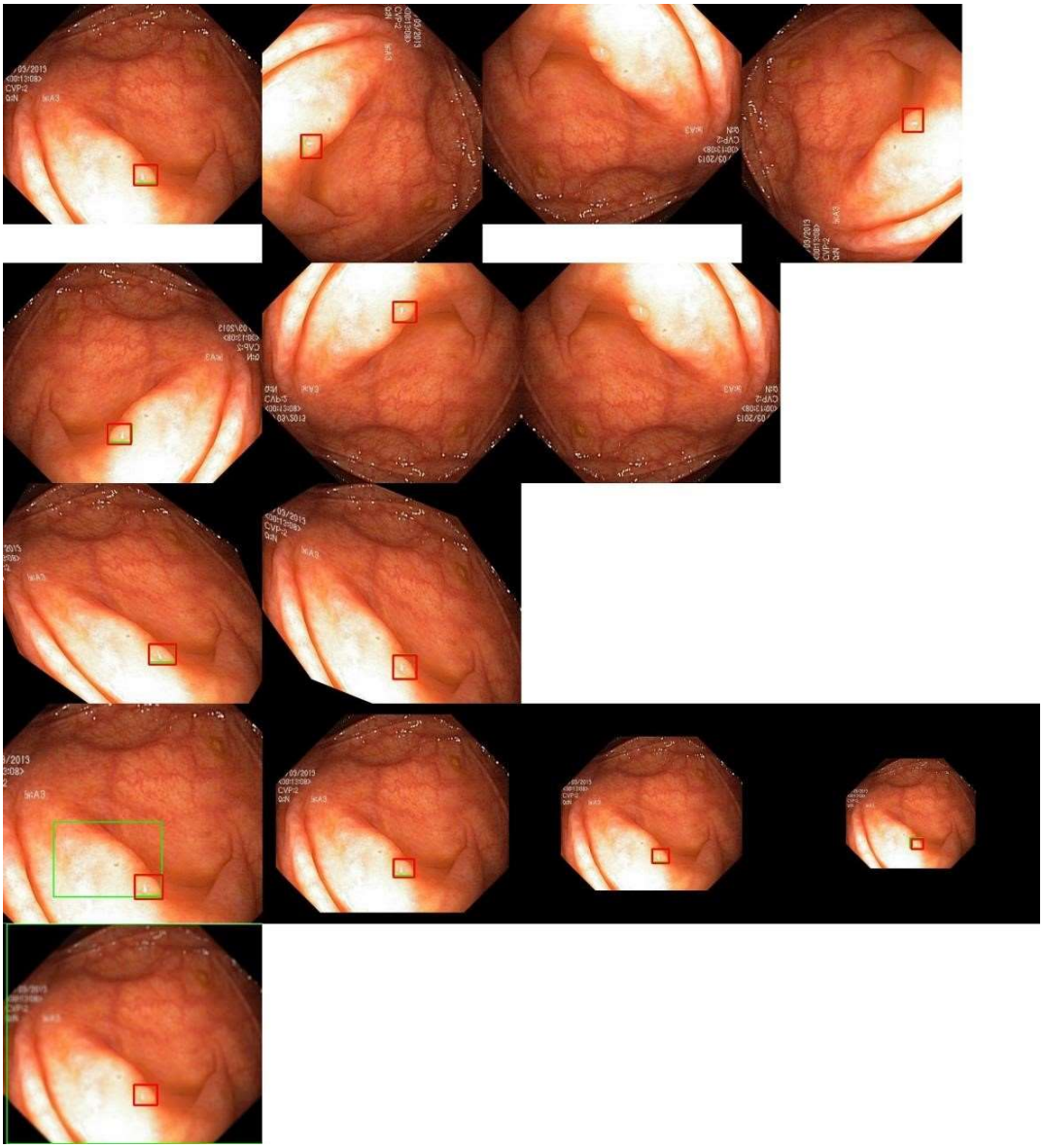

**Fig. 4.14**: This figure shows that our model is able to detect small object.

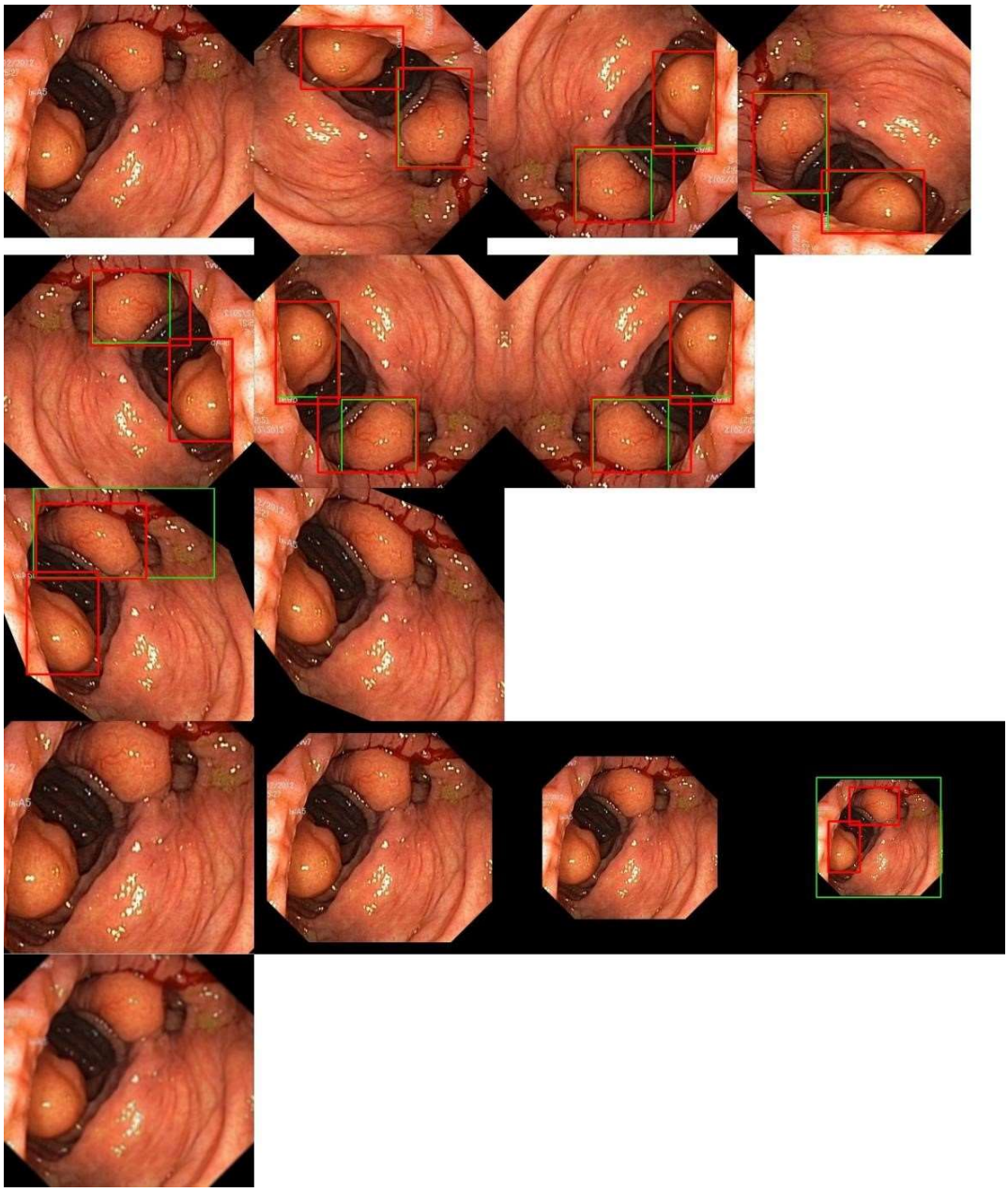

**Fig. 4.15**: This figure shows that our model is able to detect multiple object as well.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

In this study, we have proposed our model Deformable DETR with new loss function for poly detection and localization task. Our model gives better average precision as compared to stateof-the-art method as well as the existing Deformable DETR. We had done the ultimate test analysis that is TTA(test time augmentation) that helped us to improve our model performance. Our model is also giving reasonable performance in case of small objects.

In future we can further study about Deformable DETR to improve the results and one improvement of DETR has already launched called DAB-DETR[ZLL+22] (Dynamic Anchor Box Detection transformer) which can give better speed and accuracy as compared to Deformable DETR. Currently DAB-DETR got first rank among all the detectors [https://paperswithcode.com/sota/object-detection-on-coco]. We can also perform crossfolding (i.e. training and testing on different datasets) to check the performance of our model.

# REFERENCES

**[AJpt]** Mattsson F Lagergren J. Asplund J, Kauppila JH. Survival trends in gastric adenocarcinoma: A population-based study in sweden. pubmed.gov, pages 2693–2702, 2018 sept.

**[Alian]** Zhou F. Braden B. et al. An Ali, S. objective comparison of detection and segmentation algorithms for artefacts in clinical endoscopy. nature.com, 2020 Jan.

**[AWP+09]** Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. Texture-based polyp detection in colonoscopy. In Hans-Peter Meinzer, Thomas Martin Deserno, Heinz Handels, and Thomas Tolxdorff, editors, Bildverarbeitung f¨ur die Medizin 2009, pages 346–350, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

**[BTS+17]** Jorge Bernal, Nima Tajkbaksh, Francisco Javier S´anchez, Bogdan J. Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, Konstantin Pogorelov, Sungbin Choi, Quentin Debard, Lena Maier-Hein, Stefanie Speidel, Danail Stoyanov, Patrick Brandao, Henry C´ordova, Cristina S´anchezMontes, Suryakanth R. Gurudu, Gloria Fern´andez-Esparrach, Xavier Dray, Jianming Liang, and Aymeric Histace. Comparative validation of polyp detection methods in video colonoscopy: Results from the mic39 cai 2015 endoscopic vision challenge. IEEE Transactions on Medical Imaging, 36(6):1231–1249, 2017.

**[BWL20]** Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

**[CMS+20]** Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

**[DBK+20]** Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

**[GPAM+14]** Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David WardeFarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

**[HR01]** editors. Holzheimer RG, Mannick JA. Surgical treatment: Evidencebased and problem-oriented. pubmed.gov, 2001.

**[HW19]** Zhanchao Huang and Jianlin Wang. Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection, 2019.

**[HZRS15]** Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

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

**[J.ul]** Lee J. Resection of diminutive and small colorectal polyps: What is the optimal technique? pubmed.gov, pages 355–8, 2016 jul.

**[JAT+21]** Debesh Jha, Sharib Ali, Nikhil Kumar Tomar, Havard D. Johansen, Dag Johansen, Jens Rittscher, Michael A. Riegler, and Pal Halvorsen. Real-time polyp detection, localization

and segmentation in colonoscopy using deep learning. IEEE Access, 9:40496–40510, 2021.

**[JSR+19a]** Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, P˚al Halvorsen, Thomas de Lange, Dag Johansen, and H˚avard D. Johansen. Kvasir-seg: A segmented polyp dataset, 2019.

**[JSR+19b]** Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Dag Johansen, Thomas de Lange, Pal Halvorsen, and Havard D. Johansen. Resunet++: An advanced architecture for medical image segmentation, 2019.

**[KSep]** Maroulis DE Karras DA Tzivras M. Karkanis SA, Iakovidis DK. Computer-aided tumor detection in endoscopic video using color wavelet features. pubmed.gov, pages 141–52, 2003 Sep.

**[Leepr]** Jeong J. Song E.M. et al. Lee, J.Y. Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets. nature.com, 2020 Apr.

**[QSS+21]** Hemin Ali Qadir, Younghak Shin, Johannes Solhusvik, Jacob Bergsland, Lars Aabakken, and Ilangko Balasingham. Toward real-time polyp detection using fully cnns for 2d gaussian shapes prediction. Medical Image Analysis, 68:101897, 2021.

**[RDGF15]** Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015.

**[RF16]** Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

**[RHGS15]** Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster rcnn: Towards real-time object detection with region proposal networks, 2015.

**[SHeb]** Gao M Lu L Xu Z Nogues I Yao J Mollura D Summers RM. Shin HC, Roth HR. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. pubmed.gov, pages 1285–98, 2016 Feb.

**[SLZ21]** Zhiqiang Shen, Chaonan Lin, and Shaohua Zheng. Cotr: Convolution in transformer network for end to end polyp detection, 2021.

**[SQA+18]** 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.

**[SQB18]** Younghak Shin, Hemin Ali Qadir, and Ilangko Balasingham. Abnormal colon polyp image synthesis using conditional adversarial networks for improved detection performance. IEEE Access, 6:56007–56017, 2018.

**[TSG+16]** 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 Transactions on Medical Imaging, 35(5):1299–1312, may 2016.

**[WYpr]** Wong J Oh JH de Groen PC. Wang Y, Tavanapong W. Polyp-alert: near real-time feedback during colonoscopy. pubmed.gov, 2015 Apr. 42

**[Yampt]** Saito Y. Imaoka H. et al. Yamada, M. Development of a real-time endoscopic image diagnosis support system using deep learning technology in colonoscopy. nature.com, 2019 Sept.

**[ZLL+22]** Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022.

**[ZSL+20]** Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection, 2020.

**[ZWL+19]** Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression, 2019.