Development of a Real Time Endoscopic Image Diagnostic System using Deep Neural Network

Exam roll: 14611

Registration No. : 2016-414-697

Exam roll: 14610

Registration No. : 2016-414-705

Session: 2016–2017

Course: RME 410: Project/Dissertation

A Project submitted for the degree of
B.Sc. in Robotics and Mechatronics Engineering

Department of Robotics and Mechatronics Engineering
University of Dhaka, Dhaka-1000, Bangladesh

January 2022

Declaration of Authorship

I declare that this project titled, 'Development of a Real Time Endoscopic Image Diagnostic System using Deep Neural Network' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely my own work.
- I have acknowledged all main sources of help.
- Where the project is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Countersigned: _____

Supervisor

Signed:

Candidate 01

Signed:

Candidate 02

Abstract

Human beings have been suffering from gastrointestinal diseases since time immemorial. While most of these diseases are relatively harmless and cause discomfort at best, some diseases can pose a serious threat. The gastrointestinal tract is home to some of the most commonly occuring and fatal cancers worldwide including Colorectal cancer, Gastric cancer, Esophagal cancer and so on. While no direct cure exists, it is certainly possible to prevent them if diagnosed early on. Endoscopy is the most commonly performed diagnostic procedure for gastrointestinal diseases. One of the key objectives in the development of science and technology has been to eliminate diseases, reduce sufferings and extend the lifespan of human beings. The goal of this project is to develop a deep learning based endoscopy diagnostic system capable of operating in real-time. To find a suitable deep learning model, a benchmark of a few state-of-the-art object detection models have been performed where YOLOv5 was identified as the best performing one. A few modifications are proposed on the YOLOv5 architecture to improve its performance. As the performance of deep learning models depend largely on the availability of training data, a new colonoscopy dataset was developed. Hyperparameter optimization is performed to obtain the best set of hyperparameters for the proposed models. The proposed system is able to outperform existing state-of-the-art colorectal polyp detection models achieving a higher Precision, Recall and F1 score. Additionally, a small and lightweight custom YOLOv5s model is able to operate as fast as 588.23 frames per second. In addition to colorectal polyp detection, the system is also capable of identifying potential adenomateous polyps where biopsies should be performed. For upper endoscopy images, the system is able to diagnose gastric polyps and outperform existing similar state-of-the-art models in terms of different evaluation metrices. A Graphical User Interface (GUI) has also been developed for ease of use of the proposed system.

Acknowledgements

The authors would like to acknowledge Dr. Masfique Ahmed Bhuiyan, MBBS(DMC), FCPS(Surgery), Laparoscopic and Intervention Endoscopic Surgeon at the Department of Surgery, Dhaka Medical College and Hospital for providing colonoscopy data that proved crucial for developing the colorectal polyp detection system. The authors would like to further thank him for his continued support and consultancy throughout the length of the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem definition

Gastrointestinal(GI) diseases are one of the most common disorders in human beings. A study[17] conducted among 3000 participants in a district in Bangladesh has found that 75.8% of the participants had suffered from at least one gastrointestinal disease in the past 3 months and 56.8% had suffered from three or more symptoms. While most GI diseases do not pose any serious threat, some GI diseases can be life-threatening. Figure 1.1 demonstrates the distribution of different types of cancers worldwide. Colorectal cancer(CRC) is the third most commonly occurring and the second leading cause for cancer related deaths worldwide[1]. In the United States, around 140,000 people are diagnosed with colon cancer every year and around 50,000 die from the disease every year [18]. Stomach or gastric cancer is another one which is the fifth most commonly occurring cancer and the third leading cause for cancer related deaths around the world [1]. While the mortality rate is high, an early diagnosis can greatly increase the lifespan and quality of life of an individual.

One of the most common steps of diagnosis of gastrointestinal symptoms is an endoscopy. Endoscopy is a process where a long narrow tube with a camera at the end, called an endoscope, is inserted into the body to observe the digestive tract without any surgical procedure [19]. Figure 1.2 demonstrates a few example

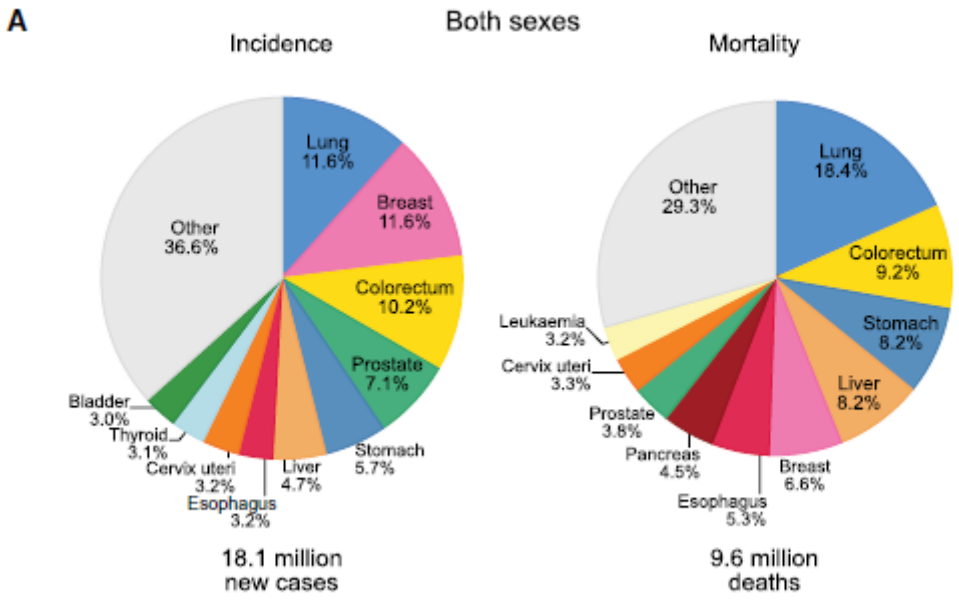


FIGURE 1.1: Incidence and mortality rate of different cancers worldwide [1]

images from endoscopy. Two types of endoscopies are more commonly performed: **esophagogastroduodenoscopy** or upper endoscopy and **colonoscopy**. In upper endoscopy, the endoscope is inserted through the mouth and used to observe the esophagus, stomach and small intestine. In colonoscopy, the endoscope (also called the colonoscope) is inserted through the anus and used to observe the colon, rectum and the large intestine.

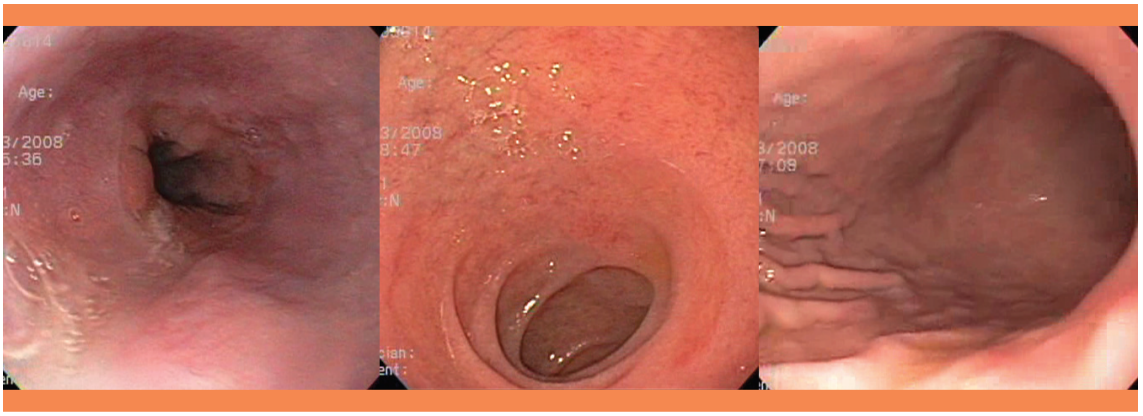


FIGURE 1.2: Sample upper endoscopy images [2]

1.2 Motivation

A study [20] has reported that undergoing colonoscopy results in a 60% reduction in CRC deaths. Another study [21] reports that a reduction of 70% in the incidence of late-stage colorectal cancer is possible by undergoing a colonoscopy. One of the most common findings during colonoscopies are polyps. Polyps are fairly common growths in the walls of the colon. While most polyps do not cause serious issues, adenomas might develop into cancers. Even if they are not cancerous, non-adenomatous polyps might cause other discomfort to individuals. Thus accurate detection of polyps is of utmost importance. However the quality of detection is greatly dependent on the endoscopist performing the procedure. One metric for determining the quality of colonoscopy is the adenoma detection rate (ADR). A study[22] has found that a 1% increase in ADR can reduce interval CRCs by about 3%. The ADR ranges from 7.4% to 52.5% based on the expertise of the operator. Hence in many cases polyps are missed which might later turn cancerous. Barret's esophagus is a metaplasia condition where the squamous epithelium cells of esophagus are changed into simple columnar epithelium cells. Barret's esophagus has a possibility of developing into adenocarcinoma, which is a form of esophagal cancer [23]. Another study [24] reports that chronic esophagitis is associated with family history of esophagal cancer. Correct diagnosis of these two conditions can thus assist in early diagnosis of esophagal cancer. The presence of tumours, ulcers and other such abnormalities in the stomach can serve as a precursor to gastric cancer. Hence correct detection of these abnormalities is necessary for an early diagnosis of gastric cancer.

In Bangladesh, the doctor-patient ratio is only 5.26 per 10,000 population. A study [25] conducted among 195 (out of a total of 197) endoscopy centres in Bangladesh reported that 38% of the centres were operated by a single endoscopist and only 9% of the centres had five or more endoscopist. Due to excessive workload and fatigue, an experienced endoscopist might miss important polyps and other such abnormalities as well. Another study [26] has indicated that the presence of an experienced nurse as a second observer can greatly increase the ADR. In absence of an experienced nurse, a computer aided diagnosis system can greatly assist the endoscopist by acting as a second observer.

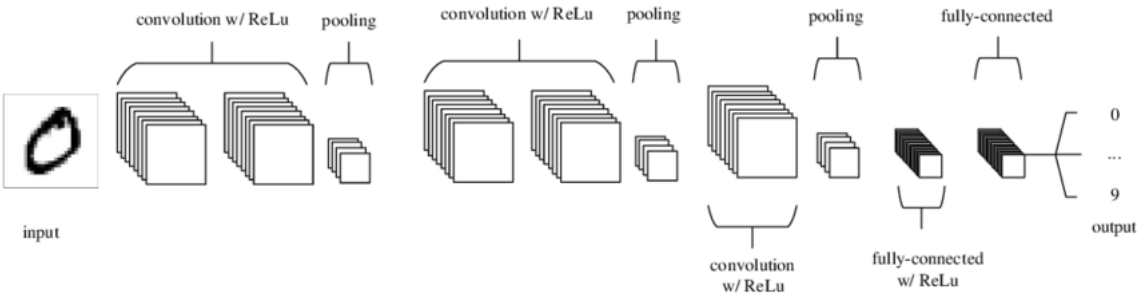


FIGURE 1.3: Architecture of a typical convolutional neural network [3]

Recently, great advancements have taken place in the field of deep neural networks. One special type of neural networks, called convolutional neural networks (CNN), are especially effective in image based applications [27]. As a result, CNNs are now successfully being used for medical image analysis. The architecture of a typical simple CNN is demonstrated in Figure 1.3. A CNN consists of multiple convolution layers, pooling layers and finally some fully connected layers. As the input image passes through these layers, useful information is extracted by the neural network which it uses to perform detection or regression tasks later on. CNNs have been used for analysing x-rays, computed tomography(CT) scans, Optical coherence tomography (OCT) scans of the retina, Magnetic Resonance Imaging (MRI) scans, Positron emission tomography (PET) scans etc. with very high accuracies comparable to that of expert physicians. The computing power of modern computers allow these tasks to be performed in real time on a live video feed. Motivated by all these factors, we propose an automated computer aided diagnosis system for endoscopic images. The system will use a deep learning based model for diagnostic purposes and will also be capable of running in real time.

1.3 Objectives

The objective of this project is to develop a complete computer aided diagnosis system that will:

- Diagnose upper endoscopic images and detect abnormalities
- Diagnose colonoscopic images and detect polyps as well as their type
- Alert the endoscopist when an abnormality is found
- Suggest whether a biopsy should be performed on the detected abnormality
- Perform aforementioned functions in real time on a live endoscopic video feed

Chapter 2

Literature Review

Recently a few works have been published on the topic of endoscopic image analysis using deep learning techniques. Some of these works are discussed briefly in this section.

2.1 Deep learning analyzes Helicobacter pylori infection by upper gastrointestinal endoscopy images

Itoh et al. [4] sought to use deep learning to diagnose Helicobacter pylori (HP) infections. The authors trained a convolutional neural network (CNN) to detect HP infections from upper gastrointestinal endoscopy images. 179 endoscopy images obtained from 139 parents were used as the dataset for the neural network. Among them, 149 images were used for the training set while the remaining 30 images were used for the test set. Using data augmentation, the training set was expanded to 596 images. The workflow of the CNNs are demonstrated in Figure 2.1 The GoogLeNet CNN pretrained for generic object detection was used and transfer learning was performed to train the network on the new dataset. The model obtained a sensitivity of 86.7% and a specificity of 86.7% with an area under the curve (AUC) of 0.956. While the system was found to perform well in terms of metrics, a few limitations are present. The neural network was trained

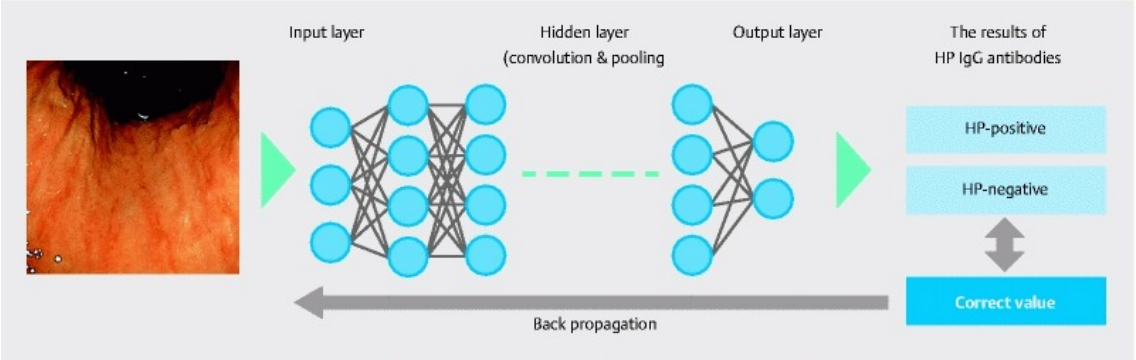


FIGURE 2.1: Workflow of the CNN presented in [4]

using a very small dataset. Hence the problem of overfitting might be present and the system might not work well on general data.

2.2 Automatic detection of erosions and ulcerations in wireless capsule endoscopy images based on a deep convolutional neural network

Aoki et al. [5] aimed to develop a deep learning based system for detecting erosions and ulcerations from wireless capsule endoscopy (WCE) images. 5360 images of small-bowel erosions and ulcerations were collected from 115 patients and used as the training dataset. The erosions and ulcerations were manually annotated by drawing a rectangular bounding box with the help of two expert endoscopists. Single Shot MultiBox Detector was used as the neural network architecture to develop the detection system. Stochastic gradient descent was performed with a learning rate of 0.0001 to train the neural network. Another 10,440 images from 65 patients were used for the validation set. The model was found to have an accuracy of 90.8%, a sensitivity of 88.2% and a specificity of 90.9%. The flowchart of the experiment is demonstrated in Figure 2.2. Although a high accuracy was obtained, the training dataset was very small which raises concerns of overfitting

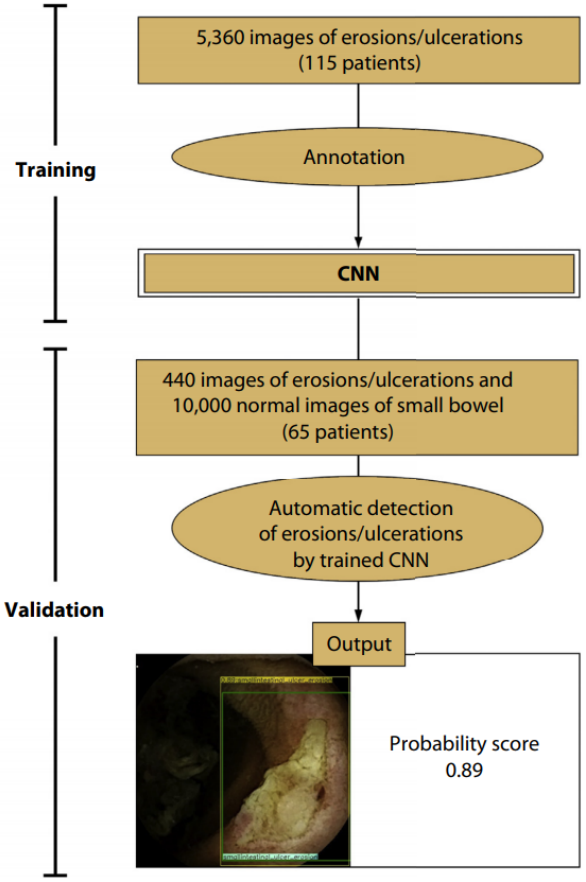


FIGURE 2.2: Flowchart of the experiment in [5]

of data. Moreover, the system was trained and tested using Pillcam WCE devices only and the effectiveness of the system for other WCE devices is unclear.

2.3 Development of a real-time endoscopic image diagnosis support system using deep learning technology in colonoscopy

Yamada et al. [6] proposed a deep learning based system for diagnosis of early signs of colorectal cancer during colonoscopy. The workflow of the system is shown in Figure 2.3. Three groups of images were used for training: the first group consisted of 1,244 still images of 1,379 polypoid lesions, the second group consisted of 2,891 images of 173 consecutive lesions and 134,983 images of noncancerous tissue obtained from videos and the third group consisted of 2,843 images of 564

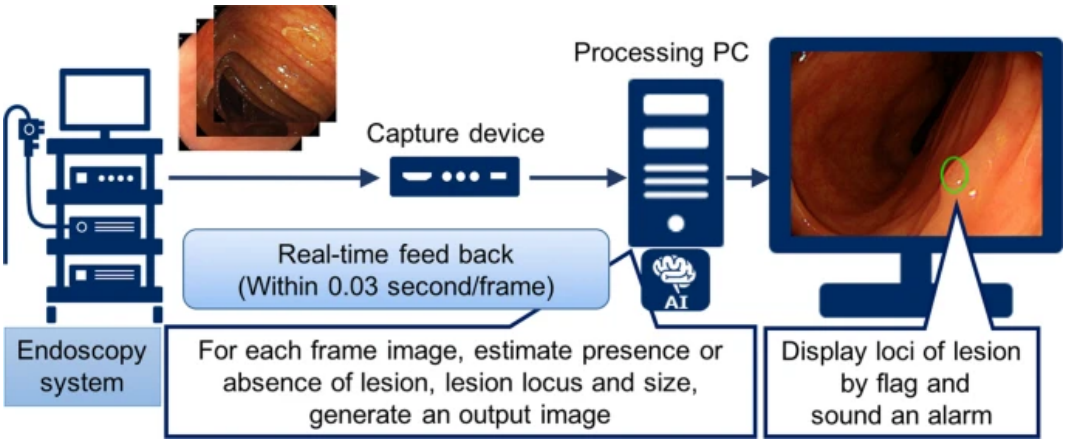


FIGURE 2.3: Workflow of the developed system in [6]

slightly elevated and depressed lesions. The VGG16 network pretrained on ImageNet weights was used and trained on the specified dataset. The system consists of two separate detection models: one for detecting lesions and another for predicting the location of lesions. In order to speed up detection, tensor decomposition was performed which resulted in the prediction speed being increased 1.7 times. The system was evaluated on a validation set consisting of 705 images of 752 lesions and 4,135 images of noncancerous tissues. The system was found to have a validation sensitivity of 97.3% and a validation specificity of 99.0%. The fast detection speed allows the system to be used in real time applications.

2.4 Real-time artificial intelligence for detection of upper gastrointestinal cancer by endoscopy: a multicentre, case-control, diagnostic study

Luo et al. [7] aimed to develop an artificial intelligence based diagnostic system for upper gastrointestinal cancers from endoscopy images which they named Gastrointestinal Artificial Intelligence Diagnostic System (GRAIDS). A dataset consisting of 1,036,496 endoscopy images collected from 84,424 patients was used for training and testing the system. Deeplab's V3+ was used as the neural network architecture for the model. The model comprises of two outputs: one output denotes whether a tumour is present or not while another output marks the region of the

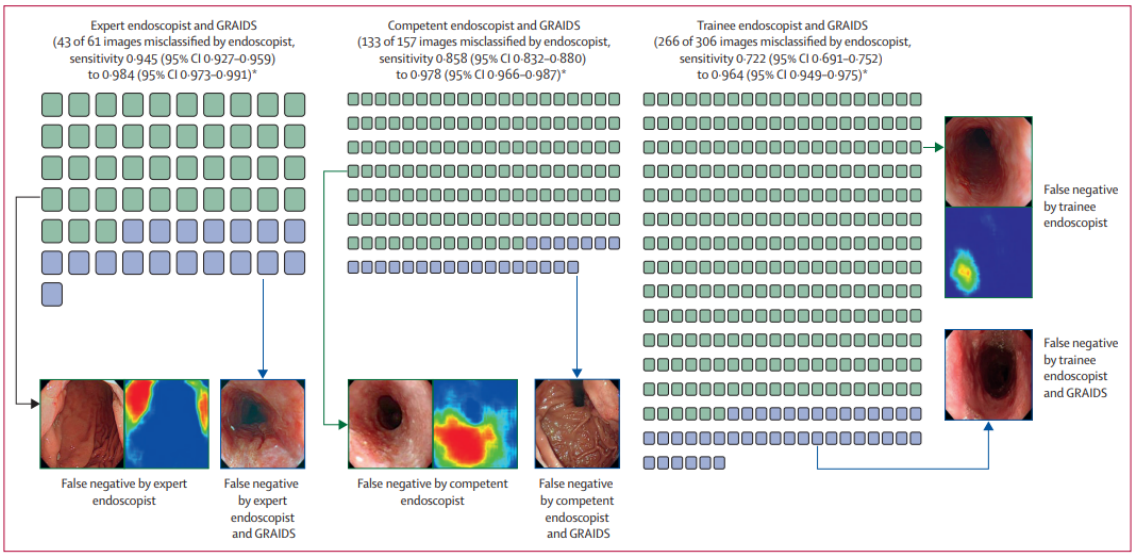


FIGURE 2.4: Performance of the GRAIDS system vs real human endoscopists [7]

tumour. After training the model achieved an accuracy in the range of 0.915 to 0.977 in a number of validation sets. The performance of the system is presented in Figure 2.4.

2.5 Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model

Byrne et al. [8] developed a deep learning based model for real time assessment of colorectal polyps from endoscopy videos. The authors followed the NICE Classification system to classify hyperplastic polyps (NICE type 1) and adenomas (NICE type 2). Videos of colorectal polyps and videos of normal mucosa were used to train the deep learning model which is based on the Inception architecture. The training dataset consisted of 223 polyp videos of which 29% were NICE type 1, 53% NICE type 2 and 18% were normal mucosa. The detection model achieved an accuracy of 94%, sensitivity of 98% and specificity of 83% when tested on a

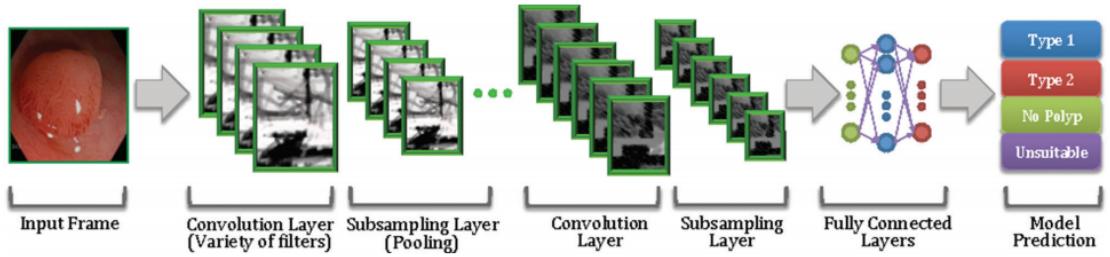


FIGURE 2.5: Schematic of the Neural Network presented in [8]

dataset of 40 videos independent from the training dataset. A schematic of the neural network is presented in Figure 2.5. Although high accuracies were obtained, the study has a few limitations. All videos used in the work were captured using the same colonoscope and so the effectiveness of the system for other models and brands of conoloscopes is unclear. The system was also evaluated on a prerecorded video so the performance of the system in real time colonoscopies is also uncertain.

2.6 Deep Learning Localizes and Identifies Polyps in Real Time With 96% Accuracy in Screening Colonoscopy

Urban et al. [9] developed a deep learning based system for detecting polyps during colonoscopies in real time. The authors experimented with different neural network architectures such as YOLO, VGG16. VGG19, ResNet50 and others. In order to develop the detection model, the authors first pretrained the neural networks on the ImageNet dataset. The training dataset consisted of two different sets: a set of 8641 colonoscopy images with 4088 unique polyp images and 4553 images without polyps and a set of 9 colonoscopy videos. To evaluate the performance of the system two more sets were used: a separate set of 1330 images with 672 unique polyp images and 658 non polyp images and a set of 11 videos which contained deliberate instances of missed polyps. Best results were obtained using the VGG19 model with an accuracy of 96.4%, AUC of 0.991 and a sensitivity of 96.9%. The performance of the architectures are presented in Figure 2.6.

Model	Initial weights	Accuracy	AUC	Sensitivity at 5% FNR	Sensitivity at 1% FNR
NPI-CNN1	—	91.9 ± 0.2%	0.970 ± 0.002	88.1%	65.4%
NPI-CNN2	—	91.0 ± 0.4%	0.966 ± 0.002	86.2%	60.6%
PI-CNN1	VGG16	95.9 ± 0.3%	0.990 ± 0.001	96.9%	87.8%
PI-CNN2	VGG19	96.4 ± 0.3%	0.991 ± 0.001	96.9%	88.1%
PI-CNN3	ResNet50	96.1 ± 0.1%	0.990 ± 0.001	96.8%	88.0%

FIGURE 2.6: Performance of different architectures used in the experiment [9]

2.7 Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images

Hirasawa et al. [28] developed an artificial intelligence based diagnostic system for detecting gastric cancer from endoscopic images. A neural network based on the Single Shot Multibox Detector architecture was used for developing the detection model. The training dataset consisted of 13,584 images of 2639 gastric cancer lesions. In order to evaluate the system, a test set consisting of 2296 images of 77 gastric cancer lesions collected from 69 patients was used. Stochastic gradient descent was used to train the network with a learning rate of 0.0001. The model achieved a sensitivity of 92.2%. Despite this high sensitivity, the study has a few limitations. The authors did not provide any information regarding the accuracy of the model. The training and test dataset were marked by a single endoscopist and so there is a possibility of misdiagnosis in the datasets. All images of the test dataset were taken using the same endoscope and so the effectiveness of the system for other endoscopes is unclear. As the system was tested using images, the effectiveness of the system for videos is uncertain as well.

2.8 Automated Polyp Detection System in Colonoscopy Using Deep Learning and Image Processing Techniques

Deep learning and image processing techniques have been employed by Kopelman et al. [10] to develop a system for detection of polyps during colonoscopy, which has been named the Automatic Polyp Detection System (APDS). The VGG16 network with some modifications was used as the neural network architecture for the system. The dataset used consisted of 120 half-minute videos of which 75 were used for training, 10 for validation and the remaining 35 for testing the system. After training and testing, the system achieved a sensitivity of 89% and a specificity of 98.4%. However the system was trained on a small dataset and hence the problem of overfitting might be present. The APDS is shown in action in Figure 2.7.

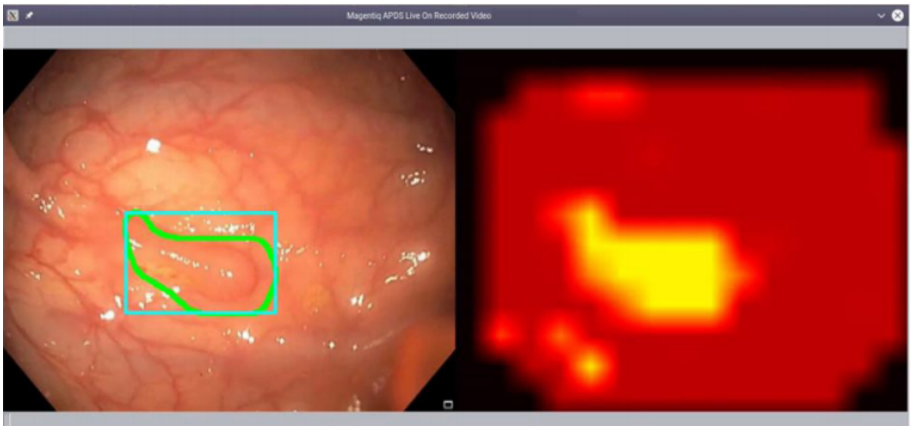


FIGURE 2.7: Display of the APDS prototype [10]

2.9 Computer-aided diagnosis of colorectal polyps using linked color imaging colonoscopy to predict histology

Min et al. [11] developed a computer-aided diagnosis (CAD) system for diagnosing colorectal polyps in linked color imaging (LCI) colonoscopy images. The detection

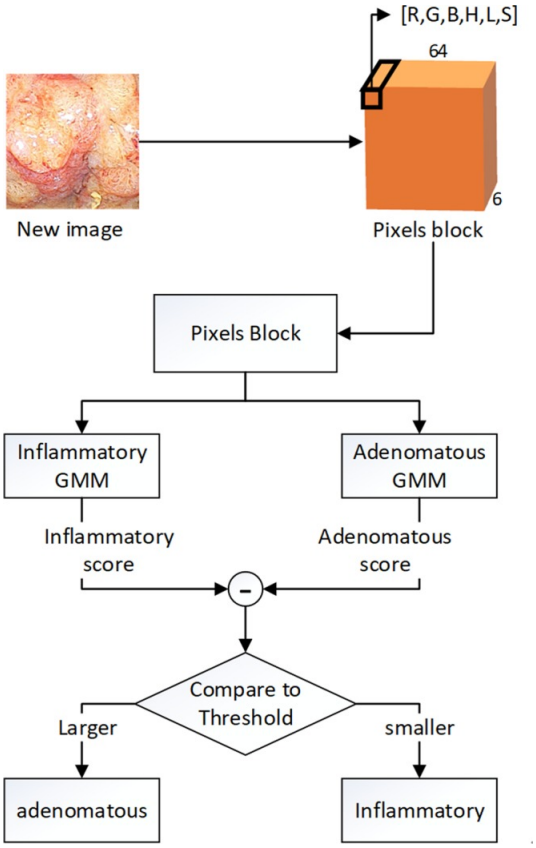


FIGURE 2.8: Flowchart of the test process [11]

system was based on the Gaussian mixture model(GMM). The system classifies polyps as adenomatous or non-adenomatous. The training dataset consisted of 139 images of adenomatous polyps and 69 images of non-adenomatous polyps. The test dataset consisted of 115 images of adenomatous polyps and 66 images of non-adenomatous polyps.The test dataset was also tested with an expert and a novice endoscopist. The test process is shown in Figure 2.8. The CAD system achieved a test accuracy of 78.4%, which was higher than that of the novice endoscopist (70.7%). However it was similar to that of an expert endoscopist(79.6%). The training dataset was very small which resulted in low performance. The system did not employ any deep learning based techniques, which have been found to perform well for image analysis. While the accuracy is close to that of an expert endoscopist, the accuracy is still low compared to other deep learning based techniques.

2.10 Toward real-time polyp detection using fully CNNs for 2D Gaussian shapes prediction

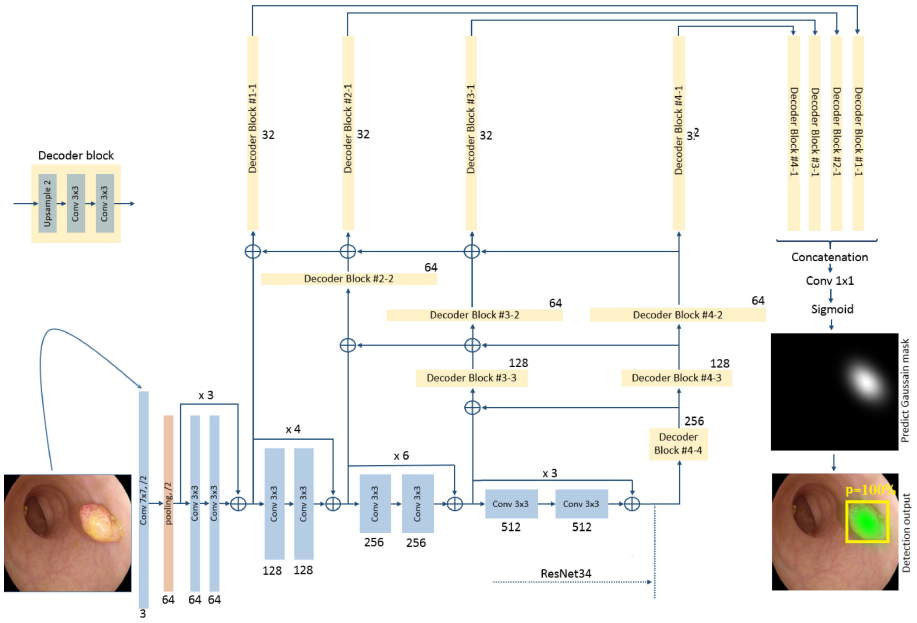


FIGURE 2.9: Real time polyp detection using MDeNetplus model [12]

Qadir et al. [12] proposed a real-time detection system for colorectal polyps. The authors developed a fully convolutional neural network (F-CNN) to segment polyps during colonoscopies. The authors postulated that binary masks typically used in segmentation datasets causes the model to focus strongly on edges resulting in false positives. To mitigate this issue they proposed the usage of Gaussian masks instead of binary masks to smoothen out the edges and reduce false positives. Three publicly available datasets were used: ETIS-LARIB, CVC-ColonDB and the CVC-ClinicDB dataset. The proposed MDeNetplus model, demonstrated in Figure 2.9, achieved a precision of 88.35% and a recall of 91%. On a 1080ti, the model requires 39ms for inference, thereby operating at a framerate of 25.6 FPS.

2.11 Stratification of gastric cancer risk using a deep neural network

Nakahira et al. [29] the authors classified the risk of individuals of gastric cancer by analysing endoscopic images using a deep neural network. The system classified the images into three classes: high risk group consisting of patients with gastric cancer, moderate-risk group consisting of patients with current or past Helicobacter pylori infection or gastric atrophy and low-risk group consisting of patients with no history of H. pylori infection or gastric atrophy. The training dataset consisted of 20,960 images of high-risk, 17,404 images of moderate-risk and 68,920 images of low risk groups. A SingleShot MultiBox Detector CNN was trained using stochastic gradient descent with a learning rate of 0.0001 for 80 epochs. 12,824 images obtained from 454 patients were used for evaluating the system. The same set was also evaluated by three endoscopist and the result of the AI based system was compared. However, no information or metric regarding the accuracy or effectiveness of the system has been provided. Thus the actual performance of the system is unclear.

Chapter 3

Methodology & Results

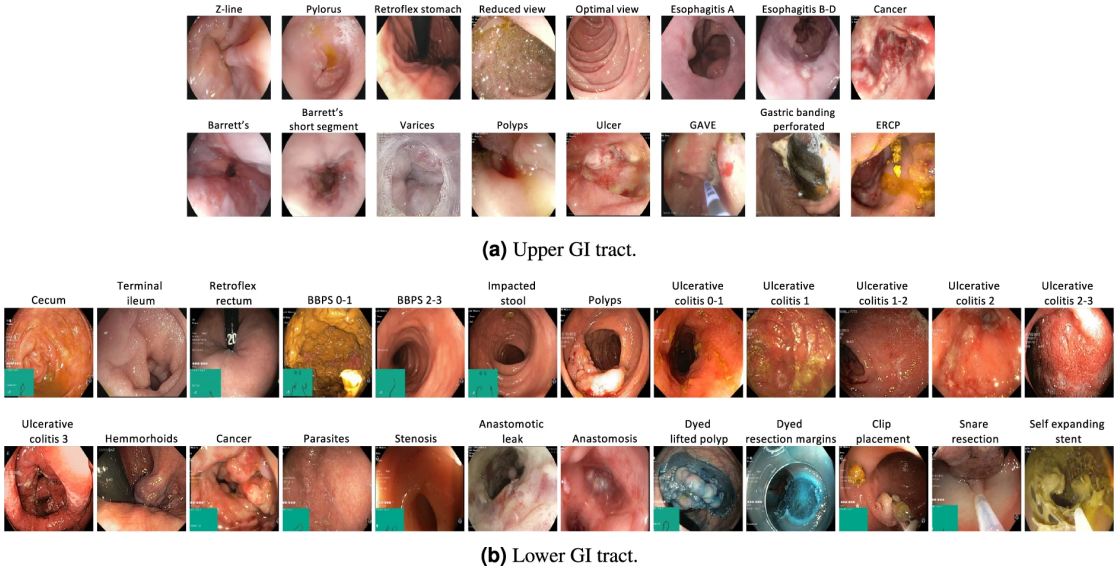
3.1 Preliminaries

3.1.1 Dataset Description

A few datasets have been collected for the purpose of training the neural networks. These datsets are briefly discussed in the following sections:

3.1.1.1 Dhaka Medical College Dataset

A new colonoscopy dataset has been developed for the purpose of the experiment. Colonoscopy data of 376 different patients were collected from the Dhaka Medical College and Hospital (DMCH) located in Dhaka, Bangladesh. A total of 20,783 images and 376 videos were collected. All colonoscopies were performed by an expert endoscopist with experience of performing over 3000 colonoscopies. Additionally with his assistance, a total of 1603 images of polyps were labelled. This dataset is referred to as the DMC dataset in the subsequent text.



(a) Upper GI tract.

(b) Lower GI tract.

FIGURE 3.1: Full list of labels in the HyperKvasir dataset [13]

3.1.1.2 The HyperKvasir dataset

HyperKvasir [13] is a large dataset consisting of 110,079 (10,662 labeled and 99,417 unlabeled) images and 374 videos of endoscopies of both upper and lower gastrointestinal (GI) tracts. The images are first divided into the upper and lower GI tract, which are subdivided into anatomical landmarks, pathological findings, quality of mucosal view and therapeutic interventions. These classes are further subdivided into more specific classes. A full list of classes and labels provided in the dataset is demonstrated in Figure 3.1.

3.1.1.3 Colonoscopy Dataset

The Colonoscopy Dataset [30] consists of 76 short videos of lesions found during colonoscopy. The dataset consists of both white light (WL) and near band imaging (NBI) videos for a total of 152 videos. The videos are classified into three groups: hyperplastic lesions, serrated lesions and adenoma lesions.

3.1.1.4 CVC-ClinicDB

The CVC-ClinicDB dataset [31] consists of 612 WL images of 31 different polyps extracted from 31 colonoscopy videos. The dataset was developed in collaboration with the Hospital Clinic of Barcelona in Spain. In addition, segmentation masks are also present for the polyps which were annotated by experts.

3.1.1.5 ETIS-Larib Polyp DB

The ETIS-Larib Polyp DB dataset consists of 196 WL images of colorectal polyps. The images are extracted from 34 colonoscopy videos and consists of images of 44 different polyps. Images are provided in 1225 $\times$ 966 resolution and segmentation masks are also provided.

3.1.1.6 Gastrointestinal Polyp Dataset

The Gastrointestinal Polyp dataset was developed by the authors in [32]. The dataset consists of 404 images of gastric polyps collected from endoscopies performed on 215 patients at the Sir Run Run Shaw Hospital in Zhejiang Province, China. Images are provided in the size of 560 x 475. Among the 404 images, 354 were augmented by rotating 180 degrees to obtain 708 images. A total of 758 images are present in the dataset.

3.1.2 Evaluation Metrics

Precision is a measure of how well the classifier is able to distinguish a particular class from the other classes. Mathematically it is the ratio between the number of True Positives and all the positive predictions (True Positives + False Positives). It is given by:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{3.1}$$

Recall is a measure of how well the classifier is able to correctly identify all occurring instances of a particular class. Mathematically it is the ratio between the number of True Positives for a particular class and the number of actual instances of that class (True Positive + False Negatives). It is given by:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3.2}$$

F1-Score is a popular metric to evaluate the performance of a classifier. It can be defined as:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.3}$$

Mean Average Precision (mAP) is the most popular evaluation metric used in object detection. Average Precision can be defined as:

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k+1) * Precisions(k)] \tag{3.4}$$

Recalls(n) = 0, Precisions(n) = 1, n = Number of thresholds

Mean average precision is obtained using:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} = AP_k \tag{3.5}$$

AP_k = AP of class k

n = Number of classes

In this experiment, the mean average precision at 0.5 intersection over union (IoU) i.e. mAP @ 0.5 IoU is used. In addition, mAP @ [0.5:0.95] IoU is also used which refers to the average mAP of different IoU thresholds starting from 0.5 to 0.95 at 0.05 step intervals. IoU is defined as the area of intersection between predicted and ground truth bounding box over the area of union between predicted and ground truth bounding box.

$$IoU = \frac{Area\ of\ intersection}{Area\ of\ union} \tag{3.6}$$

At a 0.5 IoU threshold, when the IoU of a prediction is more than 0.5, it is classified as a True Positive. If it is less than 0.5, it is classified as a False Positive. Similarly at 0.95 IoU threshold, predictions with IoU more than 0.95 are classified as True Positive while predictions with IoU less than 0.95 are classified as False Positives.

Framerate is a metric for measuring the inference speed of the model. Framerate can be defined as the number of images or video frames processed by the model in one second.

$$Framerate = \frac{1}{Inference\ time\ (in\ seconds)} \tag{3.7}$$

3.2 Benchmarking Object Detection Models on Polyp Detection Task

In order to develop a real-time polyp detection system, the detection system is required to have fast inference speeds in addition to high detection accuracy. Complex and deeper neural networks with a large number of parameters tend to have better detection performance. However, their inference times are also higher. On the other hand, simple and shallow neural networks have significantly higher inference speeds. However, their detection performances are comparatively lower. Hence, a balance is required between inference speed and performance in order to develop a detection system that is capable of operating in real time with sufficiently good performance. To this end, a number of state-of-the-art object detection models have been benchmarked on the polyp detection task. These models are discussed in details in the following sections.

3.2.1 Faster R-CNN

Faster R-CNN [14] is one of the most popular algorithms for object detection. It is the latest iteration in the Regions with CNN family after R-CNN [33] and Fast R-CNN [34] and was published in 2015. In order to perform object detection, Region based CNN algorithms employs a regional proposal network (RPN) to generate regions of interest in the image. Afterwards, an image classification model is used to perform detection. Previous iterations such as R-CNN and Fast R-CNN employed a CPU based RPN which was computationally slow. Faster R-CNN improves on this by employing a CNN as a RPN. Moreover, the RPN CNN also shares layers with the detection network. This ultimately results in significantly improved inference speeds. Figure 3.2 demonstrates the workflow of Faster R-

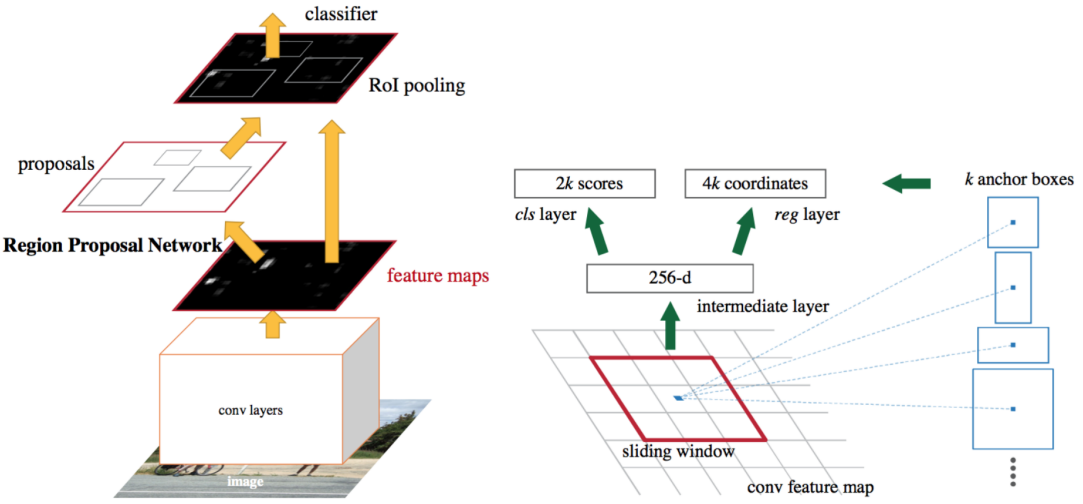


FIGURE 3.2: Object detection using Faster R-CNN [14]

CNN. The backbone CNN of Faster R-CNN can be configured to use different state-of-the-art CNNs such as VGG, ResNet, Inception etc. In this experiment, ResNeXt101 was used as the backbone network.

3.2.2 You Only Look Once (YOLO)

YOLO [15] is a state-of-the-art object detection algorithm introduced in 2016. YOLO has gained immense popularity due to its real-time inference speeds in

addition to high predictive performance. Traditional object detection algorithms employ two neural networks; a regional proposal network (RPN) that identifies regions of interest and a classification network that performs detection in the regions proposed by the RPN. Detection is performed multiple times on the same image resulting in high inference times. In YOLO, region detection and object detection are both performed in a single pass through the network. YOLO treats object detection as a regression problem where the bounding box coordinates and the objectness scores are being predicted. Figure 3.3 demonstrates the detection system of YOLO. This results in greatly improved inference times. Many iterations of YOLO has been released to further improve its performance. The original developers of YOLO continued on improving it upto version 3 which is known as YOLOv3. Afterwards, various other developers worked to further improve its performance. Versions 4 and 5 have been released afterwards, with YOLOv5 being the latest one. Besides, many other versions have also been developed by other developers such as Scaled-YOLOv4 [35] and PP-YOLO [36]. Presently, Scaled-YOLOv4 and YOLOv5 are considered the two best performing versions of YOLO with comparable performance between them as well. Hence, these two algorithms have been selected for experimentation.

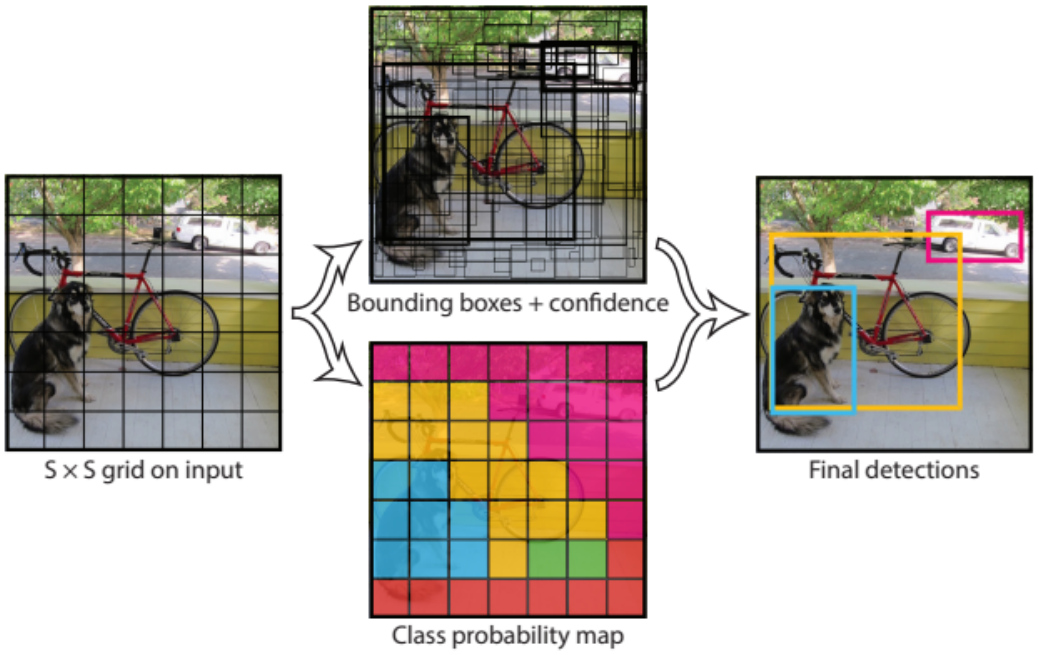


FIGURE 3.3: Object detection using YOLO algorithm [15]

3.2.3 SSD - MobileNet

A Single Shot Multibox Detector [37], or SSD in short, is an object detection algorithm that performs object detection in a single stage or single pass of the image through the network, similar to YOLO. In SSD, multiple convolution layers are appended at the end of a base network. Detection is performed on the output of each of these convolution layers, which correspond to performing detection at multiple scales. SSD detectors have very high inference speeds compared to Faster R-CNN and other multi-stage object detection models. The base network can be any CNN such as VGG, ResNets, Inception etc. and the choice of the base network impacts the performance of the detector. Research on CNNs focus on improving the predictive performance, which often results in increase of parameters. CNNs are computationally very heavy and in most cases it is impossible to obtain real time performance on mobile and embedded devices. MobileNet [38] is a lightweight architecture that has been developed to provide high inference speeds with respectable detection performance on mobile and embedded devices. An SSD detector with a MobileNet base network has been used in this experiment. The architecture of the SSD-MobileNet model is shown in Figure 3.4.

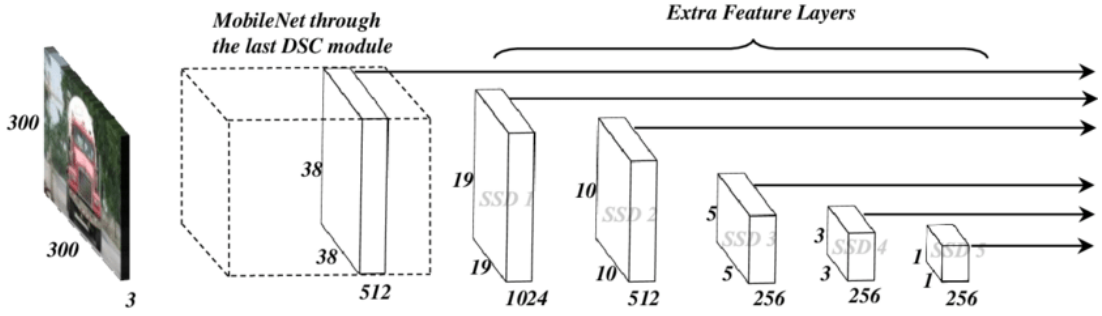


FIGURE 3.4: SSD-MobileNet architecture

3.2.4 Data Preprocessing

The HyperKvasir dataset discussed in Section 3.1.1.2 consists of 1000 labelled images of polyps with bounding boxes. The dataset is split into 70% (700 images) training data, 20% (200 images) validation data and 10% (100 images) test data. All images were resized to 416 x 416. Two augmentation operations were applied

on the training data. Random gaussian blurring between 0 to 4 pixels was applied on the images to simulate the focused and defocused conditions on the camera. In addition, the exposure was randomly varied between -25% and +25% to simulate different lighting conditions that may occur. After augmentation, 2400 training images were obtained.

3.2.5 Experimentation

All experiments were performed on a high-end GPU system and a low-end GPU system. The high-end GPU system consists of an Nvidia Tesla P100 GPU with 16GB VRAM, 16GB system RAM and a 2 core CPU. The low-end GPU system consists of an Nvidia MX330 GPU with 2GB VRAM, 8GB system RAM and an Intel Core i5-1135G7 CPU.

Four different object detection models were tested. The training configurations are specified below. No hyperparameter tuning was performed for any of the models.

YOLOv5 consists of several models ranging from a lightweight YOLOv5s model with 7.3M parameters to a heavy YOLOv5x model with 87.7M parameters. Two intermediate models, YOLOv5m and YOLOv5l are present. In this experiment, both YOLOv5s and YOLOv5x models were tested. Both models were trained for 100 epochs. YOLOv5s was trained at a batch size of 128 while YOLOv5x was trained with a batch size of 16. Due to shortage of data, transfer learning was used with weights pretrained on the MS COCO dataset.

The Scaled-YOLOv4 model was trained for 100 epochs with default hyperparameters and a batch size of 32. Transfer learning was performed using weights pretrained on the MS COCO dataset.

The YOLO algorithm contains a multiscale detection strategy termed as Test Time Augmentation (TTA). When predicting using TTA, different augmentation operations are performed on the input image and the image is evaluated at different scales and augmentations. Afterwards, the predictions are merged to obtain a single final prediction. In some scenarios, TTA might result in better performance of the model while in other cases the results might be worse. Hence, both single scale and TTA evaluation has been tested for the YOLO based models.

The SSD detector with MobileNETv2 backend was trained for 100,000 steps at a batch size of 12 and using default hyperparameters. Pretrained MS COCO weights were also used for transfer learning.

The FasterRCNN model with ResNeXt101 backend was trained for 10,000 steps at a batch size of 4 using default hyperparameters.

3.2.6 Results

The purpose of this experiment is to identify suitable models with real time inference capabilities. However, the term "real-time" itself is arbitrary and no standard value exists. Cinemas and films generally have an average framerate of 24 frames per second (FPS). Analyzing the videos in the datasets it was observed that the videos have an average framerate of 25 FPS. Hence, 25 FPS has been set as the minimum target framerate for real time performance.

Model	**mAP**	**Inference time**	**Framerate**
YOLOv5x	**0.928**	0.021s	47.62 FPS
YOLOv5x (TTA)	0.925	0.051s	19.61 FPS
YOLOv5s	0.887	**0.0027s**	**370.37 FPS**
YOLOv5s (TTA)	0.911	0.0077s	129.87 FPS
Scaled YOLOv4	0.905	0.0443s	22.57 FPS
Scaled YOLOv4 (TTA)	0.907	0.112s	8.92 FPS
SSD (MobileNET)	0.881	0.014s	71.42 FPS
FasterRCNN (ResNeXt101)	0.903	0.1972s	5.07 FPS

TABLE 3.1: Benchmark of different object detection algorithms

The performance of the models are demonstrated in Table 3.1. From the table it can be observed that YOLOv5s achieved the highest framerate. When using single scale evaluation, the model is capable of achieving a framerate of 370.37 FPS at a mAP of 0.887. Using TTA improves the mAP to 0.911 at a lower framerate of 129.87 FPS, which is still significantly higher than the threshold of 25 FPS. The next best framerate was achieved by the SSD-MobileNet model with a framerate of 71.42 FPS and a mAP of 0.881. However, the YOLOv5s performed better both in terms of framerate as well as the mAP score. The slowest model was the FasterRCNN model with a framerate of 5.07 FPS. This is well below the threshold for real time detection. Moreover, it achieved an mAP of 0.903, which is lower than

the YOLOv5s model when using TTA. The Scaled YOLOv4 achieved a framerate of 22.57 FPS and a mAP score of 0.905. Although the framerate is lower than the threshold, the difference is very small and thus it can still be used as a potential candidate. A more powerful GPU can also help achieve at least 25 FPS. When using TTA, the mAP improved slightly to 0.907. However, the framerate dropped to only 8.92 FPS. Moreover, YOLOv5s with TTA beat scaled YOLOv4 both in terms of mAP as well as FPS. The best mAP score was achieved by the YOLOv5x model. The model achieved a mAP of 0.928 at a framerate of 47.62 FPS, which is over the threshold. Contrary to the YOLOv5s and Scaled YOLOv4 models, using TTA resulted in the mAP dropping to 0.925. Additionally, the framerate also dropped to 19.61 FPS, which is below the threshold.

After analyzing the results, YOLOv5s and YOLOv5x can be identified as the two most potential candidates for real time detection. Although scaled YOLOv4 achieved competitive scores, it has been discarded as a candidate due to its low framerates.

3.3 Architectural Modifications

In order to improve the performance of the model, a few modifications have been made to the architecture of the model. These are described in the following sections.

3.3.1 Depthwise Convolution

The key ingredient of Convolutional Neural Networks (CNNs) is the convolution operation. CNNs are composed of convolution layers stacked one after another. The input to a convolutional layer is convolved using a filter/kernel and the output of the convolution operation is passed into another layer. These convolution layers operate as feature detectors extracting different important features from the input image. The earlier layers extract low level features while the later layers extract high level features. Thus when training a CNN, it is essentially learning to be a robust feature extractor. Generally convolution is performed over the entire depth

of the input. For example, if the input to a convolution layer has c_{in} channels, then a filter of size $n \times n \times c_{in}$ will be required. Some common values of n include 3, 5, 7 and such. For a convolution layer with $n = 3$ and 64 input channels, a total of $3 \times 3 \times 64 = 576$ parameters will be required ignoring the bias term. The resultant of this convolution will consist of 1 output channel only. However, it is desirable to obtain multiple output channels as well. Thus in order to obtain 128 output channels, a total of $3 \times 3 \times 64 \times 128 = 73,728$ parameters will be required. Channel sizes of 512, 1024, 2048 are very commonly used and it can be clearly observed that as the channel numbers increase, the number of parameters will also greatly increase. Moreover as multiple convolution layers are typically used, this results in greatly increased number of parameters. CNNs with millions of parameters are very commonly used and as the number of parameters increase, the inference time also increases. As the goal is to develop a real time detection system, it is necessary to reduce the number of parameters. Depthwise Convolution [39] is a type of convolution where the convolution operation is applied independently on the channels.

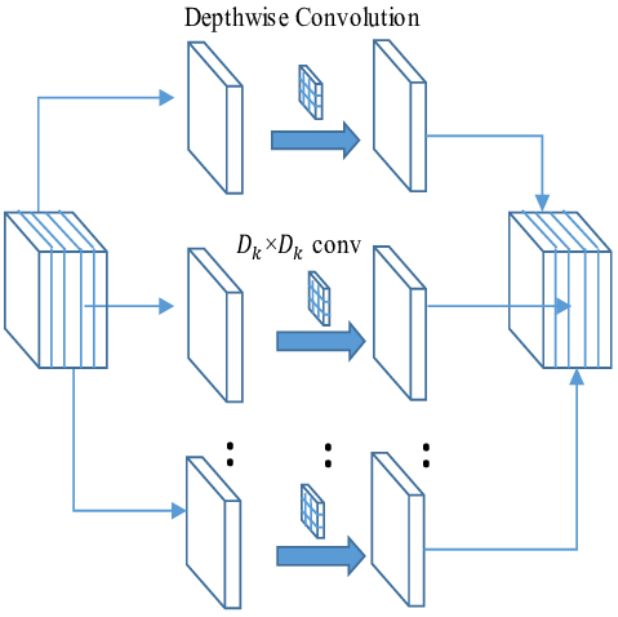
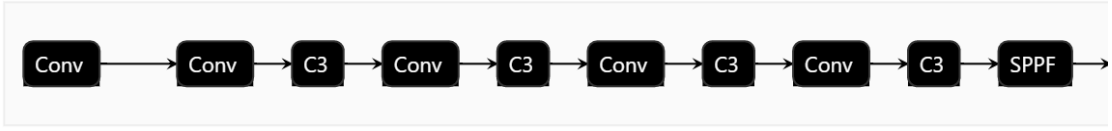


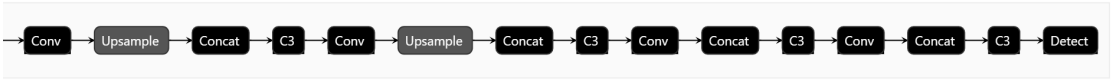
FIGURE 3.5: Depthwise convolution operation [16]

The Depthwise Convolution operation is shown in Figure 3.5. Each input channel has its own set of filters and no channel intermixing takes place. Thus for an input with c_{in} channels, a depthwise convolution layer with an $n \times n$ filter size will

consist of $n \times n \times c_{in}$ parameters. However, unlike regular convolution where a single output channel is produced, depthwise convolution will produce c_{in} output channels if 1 filter is used per input channel. Hence for a depthwise convolution layer with 64 input channels and n = 3, in order to produce 128 output channels, 2 filters will be required per input channel and a total of $3 \times 3 \times 2 \times 64 = 1152$ parameters will be required, which is significantly less than 73,728 parameters required by regular convolution.



(a)



(b)

FIGURE 3.6: (a) Backbone of the YOLOv5 network (b) Head of the YOLOv5 network

The original YOLOv5 architecture is demonstrated in Figure 3.6. The YOLOv5 architecture consists of a backbone network which is tasked with feature extraction and a head which performs detection. The backbone network as well as the head consists of Bottleneck Cross Stage Partial (CSP) Network [40] blocks (denoted using C3 in the Figure 3.6) as well as regular convolution blocks (denoted using Conv in Figure 3.6). In addition, the backbone network also consists of a Spatial Pyramid Pooling (SPP) [41] layer. The convolution blocks consist of a a regular convolution layer followed by a batch normalization layer and a SiLU activation function [42]. In order to reduce the number of parameters and thereby increase inference speeds, we propose replacing these regular convolution layer in these block with depthwise convoluion. Regular convolution has been used for the input layer and the remaining convolution blocks have been replaced with depthwise convolution blocks. The convolution layers inside the Bottleneck CSP network as well as the SPP layer is left as it is in order to ensure optimal feature extraction.

3.3.2 Transpose Convolution

The YOLOv5 architecture performs detection at three different scales of the feature map. In order to obtain these three different scales, two upsampling layers are used (denoted using Upsample in Figure 3.6). In these upsampling layers, the input feature map is upsampled to twice its height and width using nearest neighbour interpolation. However, the key issue with any upsampling algorithm is that it introduces noise. This additional noise might negatively impact the detection performance of the network. Hence, we propose the usage of transpose convolution layers [43] replacing these upsampling layers.

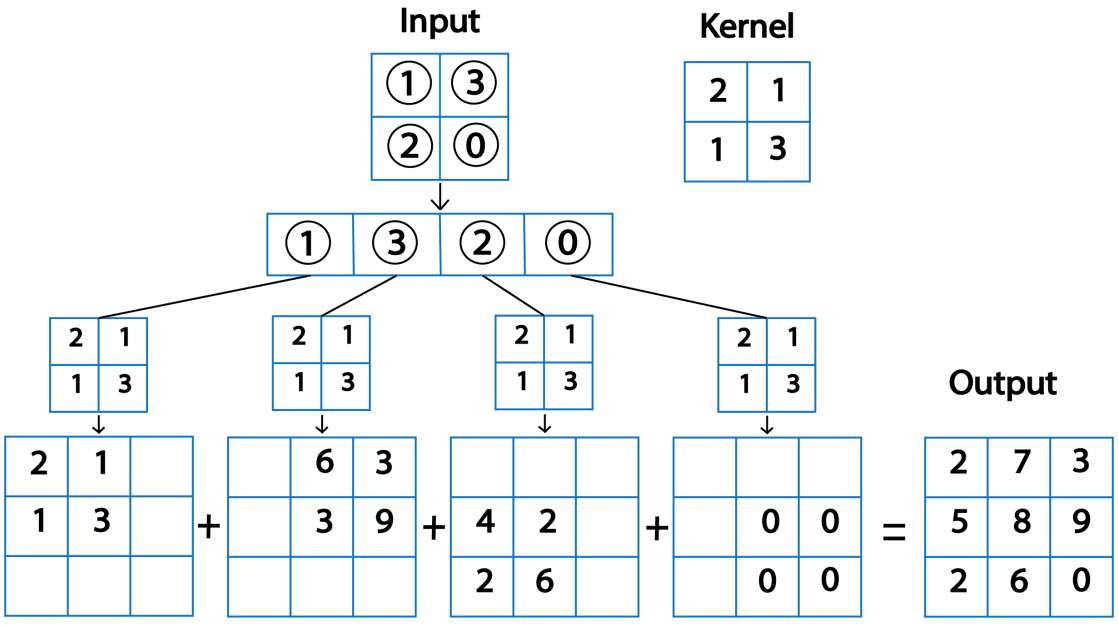


FIGURE 3.7: Transpose convolution operation

The transpose convolution operation is demonstrated in Figure 3.7. Transpose convolution can be thought of as the inverse of standard convolution. Each member of the input is multiplied with the kernel and the resultant is placed into a new matrix in such a manner that the first element of the resultant coincides with the position of the input element. After all elements of the input are multiplied, all the resultants are summed together to obtain the result of the operation. Transpose convolutions perform upsampling operations with the added benefit that they contain learnable parameters and hence they can be trained. By using transpose convolution layers, it is expected that in addition to upsampling, the feature maps

will also be enriched with valuation information that will assist in improving the detection performance of the network.

3.3.3 Single anchor box

YOLO uses the concept of anchor boxes to perform object detection. After passing through the network, an n x n grid is obtained. For each grid box, the probability of containing the centre of an object is determined along with the class and the dimensions of the bounding box. However, an image might contain multiple overlapping objects with centres lying on the same grid box. In order to handle this complication, the concept of anchor box has been introduced. For each grid, a few bounding boxes with predefined dimensions and aspect ratios are used, which are termed anchor boxes. Instead of predicting the height and width of the bounding boxes directly, the deviations from these predefined dimensions are determined. In every grid, object detection is performed for every anchor box. Afterwards, non-maxima suppression (NMS) is performed to obtain the final predictions. Thus if overlapping objects are present, they will be detection by separate anchor boxes. In YOLOv5, 3 anchor boxes are used. However, in our application, no overlapping objects of different classes are present. Hence all anchor boxes are detecting the same object and additional unnecessary processing is performed, which ultimately results in increased detection time and lower framerates. To mitigate this issue, we propose the usage of a single anchor box.

3.3.4 Half precision

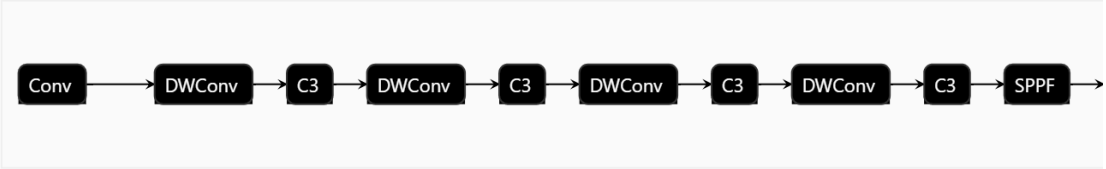
The weights, biases and different parameters of a deep learning model are generally stored using 32-bit floating point numbers. Using a higher bit number allows more precision. However, as the number of bits is increases, it also increases the memory consumption as well as computational requirements. As our goal is to improve the inference speeds, we propose storing these parameters as 16-bit floating point numbers instead of 32-bit. It is expected that this will improve the inference time with no or negligible decrease in performance.

3.3.5 Results

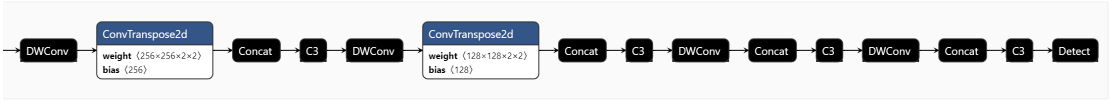
Model	**Parameters**	**mAP @ 0.5**	**mAP @ [0.5: 0.95]**	**P**	**R**	**Inference time**	**Framerate**
YOLOv5s	7.24M	0.887	0.682	0.897	0.825	0.0027s	370.37 FPS
YOLOv5s (TTA)	7.24M	0.911	0.711	0.894	0.85	0.0077s	129.87 FPS
YOLOv5s + Depthwise Convolution	4.55M	0.906	0.652	0.856	0.84	0.0022s	454.54 FPS
YOLOv5s + Depthwise Convolution + Transpose Convolution	4.87M	0.906	0.671	0.912	0.825	0.0020s	500 FPS
YOLOv5s + Depthwise Convolution + Transpose Convolution + Single Anchor	4.87M	0.908	0.683	0.894	0.89	0.0019s	526.31 FPS
YOLOv5s + Depthwise Convolution + Transpose Convolution + Single Anchor + Half Precision	4.87M	0.908	0.683	0.894	0.89	0.0017s	588.23 FPS
YOLOv5s + Depthwise Convolution + Transpose Convolution + Single Anchor (TTA)	4.87M	0.925	0.675	0.88	0.88	0.0055s	181.81 FPS
YOLOv5s + Depthwise Convolution + Transpose Convolution + Single Anchor + Half Precision (TTA)	4.87M	0.925	0.675	0.88	0.88	0.0047s	212.76 FPS
YOLOv5x	86.7M	0.928	0.736	0.895	0.895	0.021s	47.62 FPS
YOLOv5x (TTA)	86.7M	0.925	0.715	0.887	0.9	0.051s	19.61 FPS
YOLOv5x + Depthwise Convolution + Transpose Convolution + Single Anchor	72.8M	0.92	0.759	0.918	0.895	0.0146s	68.49 FPS
YOLOv5x + Depthwise Convolution + Transpose Convolution + Single Anchor + Half Precision	72.8M	0.92	0.759	0.918	0.895	0.0115s	86.95 FPS
YOLOv5x + Depthwise Convolution + Transpose Convolution + Single Anchor (TTA)	72.8M	0.934	0.763	0.912	0.885	0.0291s	34.36 FPS
YOLOv5x + Depthwise Convolution + Transpose Convolution + Single Anchor + Half Precision (TTA)	72.8M	0.933	0.763	0.912	0.885	0.0270s	37.03 FPS

TABLE 3.2: Results of different modifications made on the YOLOv5 architecture

The architecture of the proposed custom YOLOv5 model is demonstrated in Figure 3.8. Other than the input convolution block, the remaining convolution blocks have been replaced with depthwise convolution blocks. Additionally, the two upsampling layers have been replaced with transpose convolution layers and a single anchor box is used for detection.



(a)



(b)

FIGURE 3.8: (a) Backbone of the custom YOLOv5 network (b) Head of the custom YOLOv5 network

The effects of different modifications made on the YOLOv5 architecture is demonstrated in Table 3.2. For the YOLOv5s model, it can be observed that using Depthwise Convolution greatly decreases the number of parameters from 7.24M to 4.55M. As a result, the framerate is improved from 370.37 FPS to 454.54 FPS. Additionally, the mAP@0.5 and the Recall is also improved while the mAP@[0.5:0.95] experiences a drop. When using Transpose Convolution on top of this, the number of parameters slightly increases from 4.55M to 4.88M. However, it can be observed that the framerate is improved from 454.54FPS to 500FPS. It can be postulated that the nearest neighbour upsampling algorithm is computationally more expensive than the transpose convolution operation. Transpose convolution is executed faster on the GPU and hence the inference time is reduced. Moreover it can be observed that the mAP@0.5, mAP@[0.5:0.95] and the Precision is improved while the Recall is reduced. Using a Single Anchor box, the mAP, mAP@[0.5:0.95] and the Recall are improved while a slight drop in precision is observed. Additionally, the framerate is also improved to 526.31 FPS due to less computations required in

the NMS step. When operating in Half Precision mode, all performance metrics remain the same while the framerate is further improved to 588.23 FPS.

The proposed architectural modifications on the YOLOv5s model greatly improved the mAP@0.5 from 0.887 to 0.908, the Recall from 0.825 to 0.89 and the Framerate from 370.37 FPS to 588.23 FPS. The mAP@[0.5:0.95] improved slightly from 0.682 to 0.683 while the Precision experienced a very small drop of only 0.33% from 0.897 to 0.894. As our goal is to develop a real time detection system that will improve the Adenoma Detection Rate (ADR), Recall and Framerate have been identified as two important metrics, both of which have seen big improvements. For the heavier YOLOv5x model, improvements were seen as well. After the modifications, the Framerate is greatly improved from 47.62 FPS to 86.95 FPS. Additionally, the mAP@[0.5:0.95] and Precision are both improved while the Recall remain unchanged. However, the mAP@0.5 experienced a small decrease from 0.928 to 0.92. Previously when using TTA, the YOLOv5x model was operating at 19.61 FPS which is below the real time threshold requirement. However after the modifications, the framerate is improved to 37.03 FPS which is above the threshold for real time performance. Additionally, the mAP@0.5 is improved from 0.928 to 0.933, the mAP@[0.5:0.95] from 0.715 to 0.763, the Precision from 0.887 to 0.912. However, a drop of around 1.5% in Recall was observed from 0.9 to 0.885.

3.4 Colo-rectal Polyp Detection

YOLOv5s and YOLOv5x have been identified as the best performing models for polyp detection. A few modifications were made to these models to improve their performance. In order to further improve the detection performance of these networks, hyperparameter optimization has been performed. In addition to the HyperKvasir dataset, three additional datasets are used to increase the training data. This process is described in more details in the following sections.

3.4.1 Hyperparameters

When a neural network is trained, the weights are updated every iteration to better fit the dataset. However, a neural network also consists of numerous other parameters, called hyperparameters, that control the training process. These hyperparameters are set at the beginning of the training process and remain constant throughout the entire duration. The choice of hyperparameters can greatly impact the training process and overall model performance. For instance, one common hyperparameter is the learning rate, which controls the "speed" at which the network learns. If the learning rate is too high, the model might fail to converge. On the other hand if the learning rate is too low, the model might take a long time to converge.

One of the key features of the YOLO algorithm is that a number of augmentations are performed during training. Introduced in YOLOv4 [44], the authors termed these augmentations "Bag of Freebies" as these methods help in improving model performance without increasing inference time. **Mosaic augmentation** is one of the key augmentation operations performed where four different images are combined together in different aspect ratios. **MixUp** is a process where two random images are overlaid on top of each other. **HSV augmentation** involves modifying the Hue, Saturation and Value components of the images. Besides a number of **geometric augmentations** are also used such as random rotation, translation, scaling, shearing, perspective warping and flipping (both horizontal and vertical). These augmentations are controlled using separate hyperparameters that denote the probability of each individual augmentation operation happening. YOLOv5 contains 25 hyperparameters that are set at the start of the training process.

3.4.2 Hyperparameter Optimization using Genetic Algorithm

In order to find the optimal values of these hyperparameters, hyperparameter optimization is performed. In this experiment, hyperparameter optimization has been

performed using Genetic Algorithm (GA), a process termed hyperparameter evolution. Genetic algorithm is characterized by crossover and mutation operations. In GA, "parents" are first picked from a "population". A fitness function is used to assist the choice of parents. Afterwards, "crossover" takes place between the parents, where information is exchanged between the two parents and an offspring is obtained. This is followed by a "mutation" operation where the offspring is slightly manipulated to obtain the final offspring. This offspring then becomes a part of the population and the process is repeated. The entire process starting from the selection of parents to the creation of the final offspring after mutation is termed as evolution. Each such evolution constitutes a "generation".

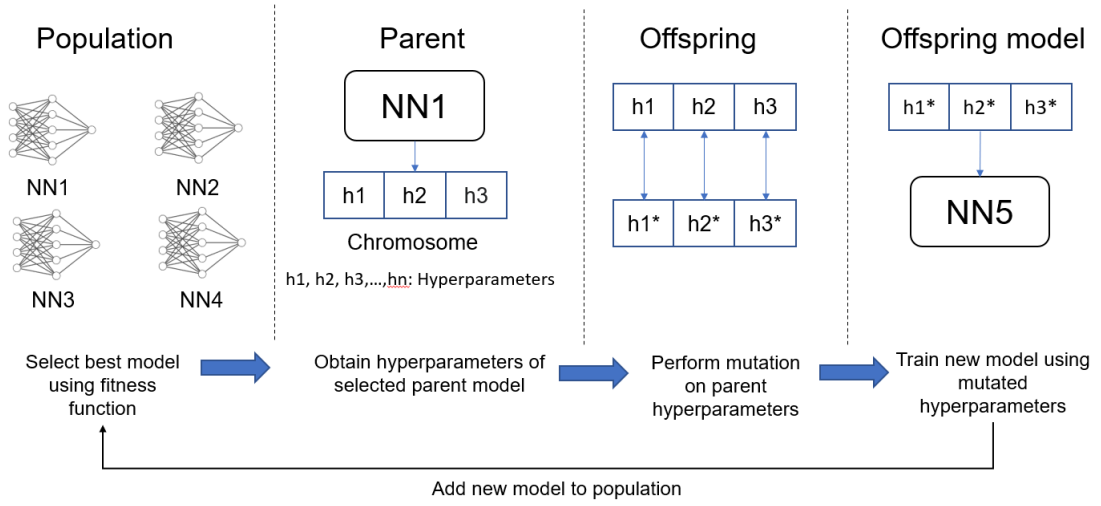


FIGURE 3.9: Hyperparameter optimization using Genetic Algorithm

The workflow of hyperparameter optimization using GA is demonstrated in Figure 3.9. In this experiment, only the mutation operation is performed. The fitness function used here is the mAP score. Given a set of hyperparameters, the respective model is trained for 10 epochs and evaluated using the fitness function. This model is then added to the population. This constitutes one evolution or generation. In the next generation, the best performing model (in terms of fitness score) is selected as a parent and its hyperparameters are mutated with a 90% probability and 0.04 variance. This process is repeated over multiple evolutions or generations until optimal results are obtained.

3.4.3 Data Preprocessing

1000 polyp images from the HyperKvasir dataset was combined with 1603 images from the DMC dataset, 612 images from the CVC-ClinicDB dataset and 196 images from the ETIS-Larib Polyp DB dataset to obtain a combined dataset of 3411 images. All images were resized to a size of 416 x 416. The dataset was divided into 70% (2388 images) training data, 20% (682 images) validation data and 10% (341 images) test data. Four augmentation operations were applied on the training set to increase the training size as well as the generalization capability of the model. The saturation, brightness and exposure were varied randomly between -25% and +25%. A random gaussian blurring between 0 to 4 pixels were also applied. 4 samples were generated per training image and a total of 9552 training images were generated.

3.4.4 Experimentation

Two experiments were performed in this section. Firstly, hyperparameter optimization was performed using the methods discussed in Section 3.4.2. Due to computational constraints, hyperparameter evolution was performed for the custom YOLOv5s model only. Hyperparameter evolution was performed for a total of 300 generations. The best hyperparameters were used to train both custom YOLOv5s and YOLOv5x models. Secondly, the models were trained using the combined dataset to further improve the performance and also to evaluate the effectiveness of using increased training data.

3.4.5 Results

The best obtained hyperparameters are presented in Table 3.3. These hyperparameters are used to train both YOLOv5s and YOLOv5x model for 100 epochs. The performance of these models are shown in Table 3.4.

When using the combined dataset, the performance of the YOLOv5x model improves to 0.946 mAP@0.5. Using TTA, the performance improves to a mAP@0.5 of 0.951. In terms of mAP@[0.5:0.95] TTA made no difference and the same score

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Initial learning rate	0.01	Final learning rate	0.2
Momentum	0.937	Weight decay	0.0005
Warmup epochs	3	Warmup momentum	0.8
Warmup initial bias learning rate	0.1	Box loss gain	0.05
Hue augmentation	0.015	Saturation augmentation	0.7
Value augmentation	0.4	Rotation	0.0 degrees
Translation	0.1 (fraction)	Scale	0.5
Shear	0 degree	Perspective	0.0
Up-down flip	0.0	Left-right flip	0.5
Mosaic augmentation	1.0	Mixup	0.0

TABLE 3.3: Best hyperparameters for YOLOv5 models

Model	**Dataset**	**mAP @ 0.5**	**mAP @ [0.5: 0.95]**	**P**	**R**	**Inference time**	**Framerate**
Custom YOLOv5x	HyperKvasir	0.92	0.759	0.918	0.895	0.0115s	86.95 FPS
Custom YOLOv5x (TTA)	HyperKvasir	0.933	0.763	0.912	0.885	0.0270s	37.03 FPS
Custom YOLOv5s	HyperKvasir	0.908	0.683	0.894	0.89	0.0017s	588.23 FPS
Custom YOLOv5s (TTA)	HyperKvasir	0.925	0.675	0.88	0.88	0.0047s	212.76 FPS
Custom YOLOv5x	Combined dataset	0.946	0.782	0.933	0.891	0.0114s	87.71 FPS FPS
Custom YOLOv5x (TTA)	Combined dataset	0.951	0.782	0.921	0.91	0.0276s	36.23 FPS
Custom YOLOv5s	Combined dataset	0.944	0.744	0.918	0.892	0.0017s	588.23 FPS
Custom YOLOv5s (TTA)	Combined dataset	0.952	0.748	0.921	0.889	0.0046s	217.39 FPS

TABLE 3.4: Evaluation metrics for models trained using best hyperparameters and combined dataset

of 0.782 was achieved in both cases. However, using TTA drops the framerate to

36.23 FPS which is still above the real time threshold after the proposed modifications have been made. For the YOLOv5s model, the combined dataset results in a mAP@0.5 of 0.944, which is marginally lower than the YOLOv5x model at more than six times the framerate. Moreover, TTA improves the performance further to a mAP@0.5 of 0.952, which is marginally better than the YOLOv5x model at six times the framerate. In terms of precision, the YOLOv5x models are found to perform better than the YOLOv5s models. The purpose of the experiment is to develop a detection system that will increase the Adenoma Detection Rate (ADR), i.e. identify the maximum number of polyps that are present, which makes the metric recall an important criteria to measure the performance of the system. The best recall was achieved by the YOLOv5x TTA model with a score of 0.91, which means that the model is able to identify 91% of all of the polyps that are present in the test set. The YOLOv5s without TTA achieved a recall of 0.892, which is slightly better than the YOLOv5x model without TTA. Additionally, the YOLOv5s TTA model achieved a recall of 0.889 which is higher than the YOLOv5x model without TTA.

Model	**P**	**R**	**F1**	**Inference time**	**Framerate**
Custom YOLOv5x (Proposed)	**0.933**	0.891	0.912	0.0114s	87.71 FPS
Custom YOLOv5x (TTA) (Proposed)	0.921	**0.91**	**0.915**	0.0276s	36.23 FPS
Custom YOLOv5s (Proposed)	0.918	0.892	0.906	**0.0017s**	**588.23 FPS**
Custom YOLOv5s (TTA) (Proposed)	0.921	0.889	0.905	0.0046s	217.39 FPS
Qadir et al. 2021 [12]	0.8835	**0.91**	0.896	0.039s	25.64 FPS
Xu et al. 2021 [45]	0.855	0.757	0.799	0.035s	28.57 FPS
Liu et al. 2021 [46]	0.778	0.875	0.824	0.108s	9.25 FPS

TABLE 3.5: Comparison of proposed model with state-of-the-art works

Table 3.5 presents a comparison of the proposed models with some other recently developed models presented in [12], [45] and [46]. All of these models were developed and tested using the publicly available datasets discussed previously with the aim of performing real-time colorectal polyp detection. The models presented in [12] and [45] were able to achieve real-time detection performance with a framerate greater than 25FPS, however they were outperformed by all of our proposed models. It can be observed that our proposed YOLOv5s and YOLOv5x models are able to outperform the state-of-the-art models in terms of precision, F1 and inference time/framerate. The model presented in [12] achieved a Recall of 0.91, which is the same as our proposed YOLOv5x model with TTA. However, our model is able to operate at a much higher framerate of 36.23 FPS.

3.5 Biopsy Suggestion

Polyps and other abnormalities can frequently develop into cancerous cells. In many cases, such adenomatous polyps can be identified easily from its gross appearance. While many abnormalities in the gastrointestinal tract can serve as precursors or signs of cancers, it is not always possible to determine from visual imagery alone whether the findings are cancerous or not. Instead, a biopsy must be performed to confirm whether cancer cells are present or not as well as to determine the actual histology of the cell. During early stages, premalignant and malignant cells are confined to small locations. Performing biopsy on cells collected from a distant location might not result in a proper diagnosis. Hence our system will also determine whether the tissue from a specific location is suitable or not and suggest the endoscopist accordingly.

3.5.1 Data Preprocessing

The Colonoscopy dataset discussed in section 3.1.1.3 was used in this experiment. The dataset consists of 76 colonoscopy videos with an average framerate of 25 frames per second. As images are required for training the models, 5 frames are extracted from every second of the videos. A total of 3330 images were extracted

from the videos. The dataset was split into 80% (2664 images) training, 10% (333 images) validation and 10% (333 images) test images. Four augmentation operations were applied on the training set. The saturation, brightness and exposure were varied randomly between -25% and +25% and random gaussian blurring between 0 to 4 pixels were applied. After augmentation, a total of 7191 training images were obtained.

3.5.2 Experimentation

From the benchmark performed in the previous sections, YOLOv5 was found to be the best performing model. Hence, only YOLOv5 was experimented with in this task and both YOLOv5s and YOLOv5x models were trained. The proposed architectural modifications were applied here as well. The models were trained to classify between hyperplastic (non-cancerous) and adenomatous (cancerous) polyps. Whenever an adenomatous polyp is detected, it will suggest the endoscopist to perform a biopsy. Due to shortage of data, transfer learning was used. Two experiments were performed in this regard. For the first experiment, transfer learning was performed using weights pretrained on the MS COCO dataset. For the second experiment, the weights from the polyp detection task was used. All models are trained for 100 epochs.

3.5.3 Results

The results of the experiments are shown in Table 3.6. A few observations can be made from the results.

Firstly, in terms of mAP, Recall and Framerate, our proposed models are outperforming their original unmodified counterparts in all of the cases. The best mAP@0.5 as well as mAP@[0.5:0.95] is achieved by our proposed YOLOv5x model when trained on MS COCO weights. The best Recall was achieved by the proposed YOLOv5s model with TTA when trained on Polyp detection task weights. However, in terms of Precision, the original unmodified models are found to perform better with the highest precision being achieved by the original YOLOv5s

Model	**Weights**	**mAP @ 0.5**	**mAP @ [0.5: 0.95]**	**P**	**R**	**Inference time**	**Framerate**
YOLOv5s	MS COCO	0.72	0.558	0.667	0.773	0.0028s	357.14 FPS
YOLOv5s (TTA)	MS COCO	0.747	0.57	0.672	0.774	0.0072s	138.88 FPS
YOLOv5x	MS COCO	0.707	0.564	0.681	0.731	0.019s	52.63 FPS
YOLOv5x (TTA)	MS COCO	0.733	0.587	0.686	0.737	0.046s	21.73 FPS
YOLOv5s	Polyp detection	0.752	0.535	**0.743**	0.766	0.0029s	344.82 FPS
YOLOv5s (TTA)	Polyp detection	0.76	0.555	0.71	0.811	0.0067s	149.25 FPS
YOLOv5x	Polyp detection	0.756	0.561	0.616	0.836	0.020s	50 FPS
YOLOv5x (TTA)	Polyp detection	0.778	0.587	0.686	0.847	0.048s	20.83 FPS
Custom YOLOv5s	MS COCO	0.806	0.59	0.626	0.893	0.0016s	625 FPS
Custom YOLOv5s (TTA)	MS COCO	0.759	0.573	0.623	0.918	0.0042s	238.09 FPS
Custom YOLOv5x	MS COCO	**0.845**	**0.672**	0.659	0.907	0.014s	71.42 FPS
Custom YOLOv5x (TTA)	MS COCO	0.814	0.65	0.641	0.917	0.028s	35.71 FPS
Custom YOLOv5s	Polyp detection	0.784	0.625	0.662	0.88	0.0018s	555.55 FPS
Custom YOLOv5s (TTA)	Polyp detection	0.778	0.602	0.676	**0.92**	0.0041s	243.90 FPS
Custom YOLOv5x	Polyp detection	0.741	0.571	0.649	0.809	0.010s	100 FPS
Custom YOLOv5x (TTA)	Polyp detection	0.781	0.605	0.682	0.856	0.028s	35.71 FPS

TABLE 3.6: Evaluation metrics of the models for the polyp classification (Biopsy suggestion) task

trained on polyp detection task weights. The biggest improvements of our modifications are seen in terms of Recall and Framerate, which is directly in line with our objective of increasing ADR and real time detection.

Secondly, when trained on MS COCO weights, in case of the original models, the smaller YOLOv5s model is found to outperform the heavier YOLOv5x model. On the other hand, using weights from the polyp detection task, the original YOLOv5x model can be seen to outperform the YOLOv5s model. Transfer learning is effective as the information learned by the previous pretrained model can transfer over to the new model and assist it in learning the new data better and quicker. The models pretrained on the MS COCO dataset learnt to differentiate between 80 different objects i.e. it contains information regarding shapes, edges, corners etc. On the other hand, the polyp detection model was trained specifically to detect and separate colorectal polyps from its surroundings. This means that the model has a much better understanding of the size, shape and structure of polyps. Hence when using this model, the networks can quickly adapt to the task of polyp classification. This is evident from the results as both YOLOv5s and YOLOv5x models achieved higher mAP when using polyp detection weights. However, when using our modified networks, the opposite was seen. Although the highest recall was achieved when trained using the polyp detection task weights, in majority of the cases, MS COCO weights were found to produce better results.

(a) Easy to identify adenomatous polyp

(b) Difficult to identify adenomatous polyp

(c) A hyperplastic polyp

FIGURE 3.10: A few images taken from the dataset

Thirdly, the performance of the models are very poor when compared to the polyp detection task. Some adenomatous polyps have a distinct appearance and they can be detected easily. However, in many cases adenomatous polyps, especially in early stages, have the same appearance as a hyperplastic polyp. In such cases, biopsies are required to ensure the cancerous nature of these polyps. Figure 3.10 demonstrates 3 images taken from the dataset. The adenomatous polyp shown in Figure 3.10(a) can easily be identified from its gross appearance. On the other

hand, the adenomatous polyp demonstrated in Figure 3.10(b) appears very similar to the hyperplastic polyp shown in Figure 3.10(c) and is misclassified as a hyperplastic polyp. It is difficult to differentiate from visual appearance alone and biopsies are usually performed in this case. However, the CNN models make predictions based on visual information alone and this results in a number of false positives and false negatives, ultimately resulting in poor mAP scores.

Finally, for all of the models, using Test Time Augmentation produced better results especially in terms of Recall, albeit at a higher computational cost. Hyperplastic polyps are generally small in size while adenomatous polyps are comparatively larger in size. This difference in size is one of the methods used to differentiate between the two visually. By evaluating the images at multiple scale using TTA, the obtains a better understanding of the size and shape of the polyps and can thus differentiate between the two types better.

3.6 Upper Endoscopy Diagnosis

Upper endoscopy covers the esophagus, stomach and duodenum. Esophagitis is an irritation or inflammation of the esophagus which might cause varying degrees of discomfort to a person. Barret's esophagus sometimes act as a precursor to esophageal cancer and thus its diagnosis is important. Various abnormalities in the stomach can also serve as early signs of gastric cancer. Similar to colorectal polyps, gastric polyps are also a common abnormality. Gastric polyps can also lead to cancer and other serious complications if left untreated. Due to shortage of training data, the system will only diagnose gastric polyps as a proof of concept.

3.6.1 Experimentation

The findings from the previous sections are incorporated in this experiment to develop a gastrointestinal polyp detection system from upper endoscopy images. Both YOLOv5s and YOLOv5x models with the proposed architectural modifications are used. The models are trained using the best hyperparameters obtained

in Table 3.3. Both models are trained for 100 epochs using a transfer learning approach on pretrained MS COCO weights.

3.6.2 Data Preprocessing

The Gastrointestinal Polyp dataset discussed in Section 3.1.1.6 was used in this experiment. The dataset consists of 758 images which was split into 80% training (606 images), 10% validation (76 images) and 10% test (76 images). In order to increase the training set size, four augmentation operations were applied. The saturation, brightness and exposure were varied randomly between -25% and +25% and random gaussian blurring between 0 to 4 pixels were applied. A total of 1818 training images were obtained after augmentation.

3.6.3 Results

Model	**mAP @ 0.5**	**mAP @ [0.5: 0.95]**	**P**	**R**	**Inference time**	**Framerate**
Custom YOLOv5x	**0.913**	**0.486**	**0.95**	0.805	0.0115s	86.95 FPS
Custom YOLOv5x (TTA)	0.908	0.478	0.891	**0.848**	0.0295s	33.89 FPS
Custom YOLOv5s	0.894	0.42	0.905	0.817	**0.0019s**	**526.31 FPS**
Custom YOLOv5s (TTA)	0.894	0.39	0.909	0.792	0.0047s	212.76 FPS

TABLE 3.7: Evaluation metrics for the Gastrointestinal polyp detection models

The performance of the gastrointestinal polyp detection models are presented in Table 3.7. It can be observed that the custom YOLOv5x model achieved better performance than the custom YOLOv5s model. In terms of mAP@0.5, mAP@[0.5:0.95] and Precision, the best performance was achieved by the custom YOLOv5x model without TTA, achieving an mAP@0.5 of 0.913, mAP@[0.5:0.95] of 0.486 and a Precision of 0.95. The custom YOLOv5x model with TTA achieved

the highest Recall of 0.845. As expected, the highest framerate was achieved by the custom YOLOv5s model, however, the custom YOLOv5x model is able to achieve 33.89 FPS with TTA and 86.95 FPS without TTA, both above the threshold for real time.

Model	**mAP @ 0.5**	**P**	**R**	**F1**	**Framerate**
Custom YOLOv5x (Proposed)	**0.913**	**0.95**	0.805	0.872	86.95 FPS
Custom YOLOv5x (TTA) (Proposed)	0.908	0.891	0.848	0.868	33.89 FPS
Custom YOLOv5s (Proposed)	0.894	0.905	0.817	0.859	**526.31 FPS**
Custom YOLOv5s (TTA) (Proposed)	0.894	0.909	0.792	0.846	212.76 FPS
Zhang et al. [32]	0.904	0.9392	0.7637	0.8424	50 FPS
Laddha et al. [47]	0.87	0.90	0.86	0.88	35 FPS
Cao et al. [48]	0.891	0.916	**0.862**	**0.888**	Not provided

TABLE 3.8: Comparison of gastrointestinal polyp detection models with existing work

Table 3.8 presents a comparison of the proposed models with a few existing state-of-the-art works in gastrointestinal polyp detection. It can be observed that our custom YOLOv5x model is able to achieve a higher mAP, Precision and Framerate than the presented state-of-the-art works. The highest Recall of 0.862 was achieved by the model presented in [48], which is about 1.6% higher than our best Recall of 0.848. However, the authors did not provide any framerate or inference speed so it is unknown whether the system is capable of operating in real time. Moreover, the authors used a private dataset of 2270 images including 1941 images of polyps while we used a small dataset of only 404 original images. Even after augmentation we had a training set of 1818 images, which is still smaller than their dataset size.

We are able to achieve a higher mAP, Precision and a comparable Recall despite using a much smaller dataset. The authors in [47] used an additional private dataset of 250 images on top of the gastrointestinal polyp dataset that was used in our experiment. However, our proposed model is able to achieve a higher mAP, Precision and Framerate and only falls behind in terms of Recall. The work presented in [32] was developed using the same dataset as us and it can be observed that our proposed custom YOLOv5x model is able to outperform them in all of the metrics mAP, Precision, Recall, F1 and Framerate. Our proposed model is already able to outperform all of the presented state-of-the-art works in terms of mAP, Precision and Framerate. Thus, given additional data, it is expected that our proposed models will be able to outperform the other two works in terms of Recall and F1 score as well.

3.7 Computer Aided Diagnosis Software

In order to deploy the trained models for Computer Aided Diagnosis (CAD), a prototype application with a Graphical User Interface (GUI) has been developed.

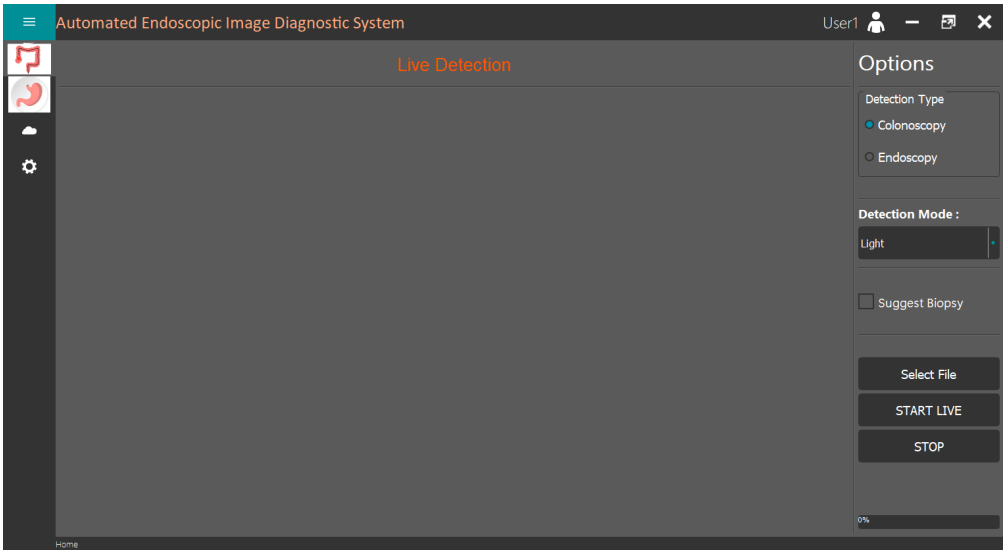


FIGURE 3.11: Home page of the application

The starting/home page of the application is shown in Figure 3.11. The application can be used to either perform detection on previously recorded images and videos or it can be used to perform detection live directly on the feed from the endoscope.

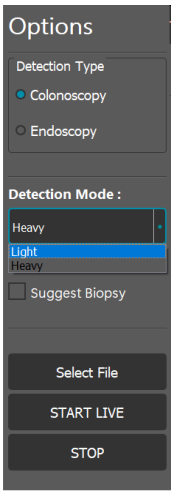


FIGURE 3.12: Enlarged view of the sidebar

On the right sidebar, a "Select File" button is present which is used to open an image or a video file for diagnosis. A "Detection Type" option is present at the top with two choices - "Colonoscopy" and "Endoscopy". As the name suggests, "Colonoscopy" mode is used for colonoscopies while "Endoscopy" mode is used for upper endoscopies. Below it a "Detection Mode" option is present with two choices - "Light" and "Heavy". When "Light" detection mode is used, the custom YOLOv5s model is used while the "Heavy" detection mode uses the custom YOLOv5x model. Figure 3.13 presents a use-case where a polyp is detected in an image using the "Heavy" detection mode.

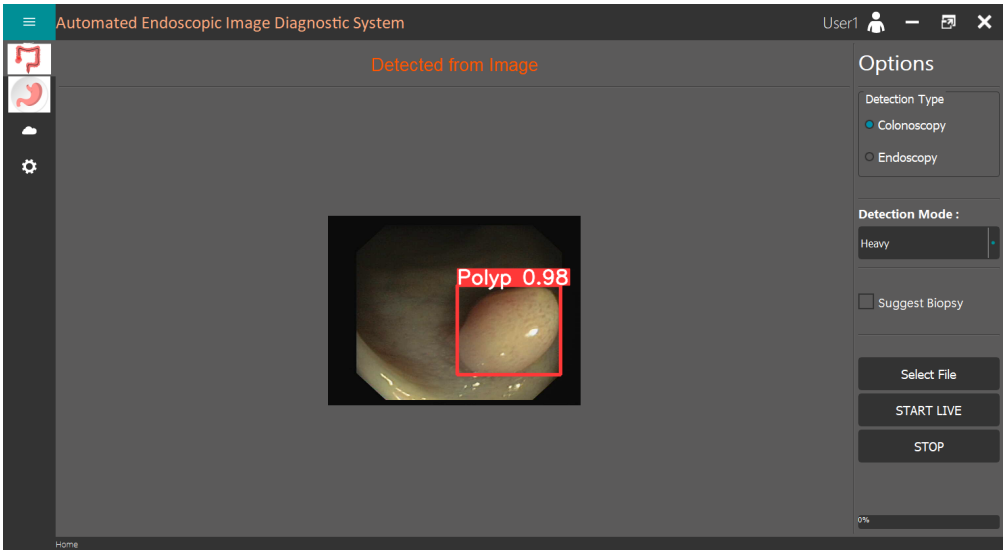


FIGURE 3.13: Detection of polyp from an image

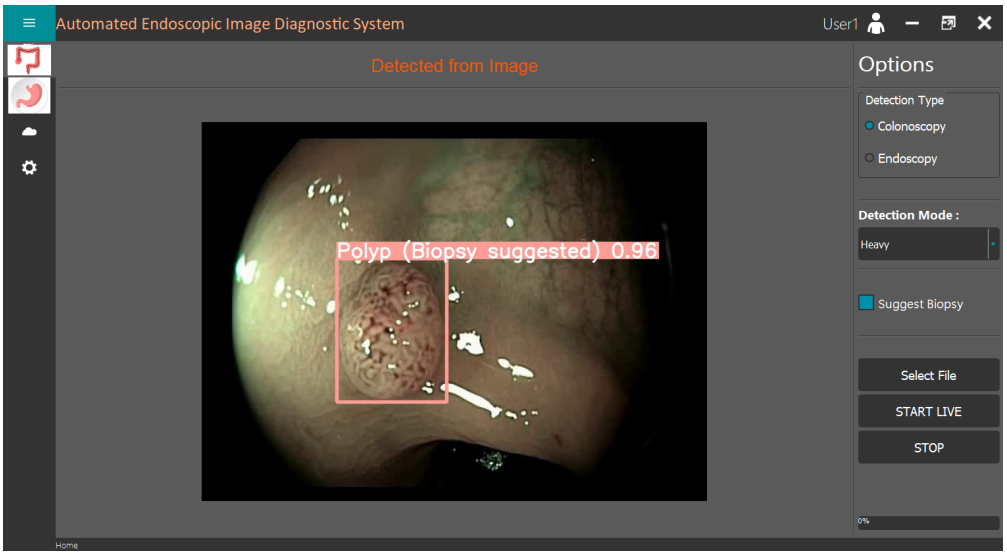


FIGURE 3.14: Biopsy suggestion on potential adenomatous polyp

The biopsy suggestion feature can be turned on or off using the "Suggest Biopsy" checkbox. By default, the option is kept off. The doctor might be interested in finding polyps only using the CAD software and not in classifying the polyp automatically. Hence, this feature has been provided as a selectable option rather than a permanent feature as it might cause irritation to the doctor. Figure 3.14 demonstrates this feature in action. An adenomatous polyp has been detected by the system and hence a biopsy is suggested to confirm the true nature of the polyp.

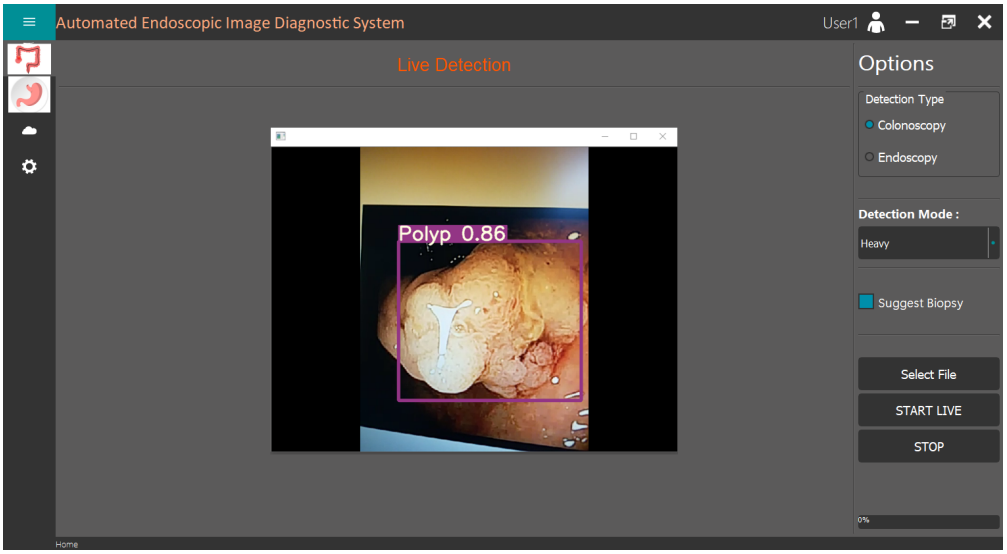


FIGURE 3.15: Polyp detection on live video feed

Pressing the "START LIVE" button in the homescreen will allow diagnosis from the camera (endoscope/colonoscope) feed directly. "Heavy" and "Light" detection modes are also present here along with a biopsy suggestion option, as like before. Pressing "STOP" will end the operation. Figure 3.15 demonstrates the operation of the model on a live video feed. In Figure 3.16, diagnosis is performed on the live video feed with the biopsy suggestion feature enabled.

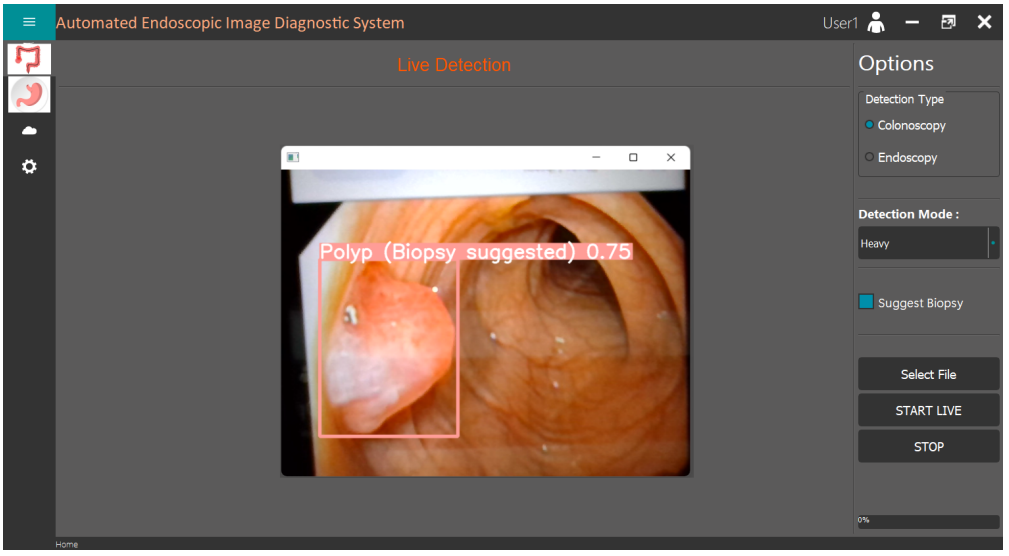


FIGURE 3.16: Biopsy suggestion on live video feed

Chapter 4

Conclusion

This project proposes a computer aided diagnosis system for endoscopy procedures. Endoscopy is one of the very first diagnostic procedures performed for diagnosing gastrointestinal problems as it allows the physician to have a direct view inside the digestive tract. Various abnormalities and lesions can be detected from an endoscopy, many of which are precursors or signs of different types of cancers of the gastrointestinal tract; colorectal cancer, gastric cancer and esophagal cancer being some of the more common ones. Detection of these abnormalities is immensely important in order to prevent or diagnose cancer at early stages. However due to various factors such as endoscopist expertise, workload, fatigue, poor image quality, poor preparation etc. many abnormalities might go undetected, which might later on prove to be life threatening for the patients. However a few studies suggest that a second observer greatly increases the detection rate of such abnormalities. Our system will detect such abnormalities in real time while an endoscopy is being performed. The system will alert the endoscopist whenever it detects such an abnormality. As the malignancy of cells cannot be determined from visual observation alone, the system will also recommend collecting tissue from a particular location for performing a biopsy if it suspects tissues of that region require further investigation. The goal of our system is not to replace endoscopists, rather it is aimed at being a complementary system to assist the endoscopists. It is expected that our system will greatly assist endoscopists in diagnosing various abnormalities in the digestive tract and thereby reduce chances of future developments of life threatening cancers.

Bibliography

[1] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 68, no. 6, pp. 394–424, 2018.

[2] "Upper endoscopy patient information from sages," Jun 2018. [Online]. Available: https://www.sages.org/publications/patient-information/patient-information-for-upper-endoscopy-from-sages/

[3] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[4] T. Itoh, H. Kawahira, H. Nakashima, and N. Yata, "Deep learning analyzes helicobacter pylori infection by upper gastrointestinal endoscopy images," *Endoscopy International Open*, vol. 06, pp. E139–E144, 02 2018.

[5] T. Aoki, A. Yamada, K. Aoyama, H. Saito, A. Tsuboi, A. Nakada, R. Niikura, M. Fujishiro, S. Oka, S. Ishihara, T. Matsuda, S. Tanaka, K. Koike, and T. Tada, "Automatic detection of erosions and ulcerations in wireless capsule endoscopy images based on a deep convolutional neural network," *Gastrointestinal Endoscopy*, vol. 89, no. 2, pp. 357 – 363.e2, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016510718332000

[6] M. Yamada, Y. Saito, H. Imaoka, M. Saiko, S. Yamada, H. Kondo, H. Takamaru, T. Sakamoto, J. Sese, A. Kuchiba *et al.*, "Development of a real-time endoscopic image diagnosis support system using deep learning technology in colonoscopy," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[7] H. Luo, G. Xu, C. Li, L. He, L. Luo, Z. Wang, B. Jing, Y. Deng, Y. Jin, Y. Li *et al.*, "Real-time artificial intelligence for detection of upper gastrointestinal cancer by endoscopy: a multicentre, case-control, diagnostic study," *The Lancet Oncology*, vol. 20, no. 12, pp. 1645–1654, 2019.

[8] M. F. Byrne, N. Chapados, F. Soudan, C. Oertel, M. L. Pérez, R. Kelly, N. Iqbal, F. Chandelier, and D. K. Rex, "Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model," *Gut*, vol. 68, no. 1, pp. 94–100, 2019.

[9] G. Urban, P. Tripathi, T. Alkayali, M. Mittal, F. Jalali, W. Karnes, and P. Baldi, "Deep learning localizes and identifies polyps in real time with 96% accuracy in screening colonoscopy," *Gastroenterology*, vol. 155, no. 4, pp. 1069–1078, 2018.

[10] Y. Kopelman, O. Gal, H. Jacob, P. Siersema, A. Cohen, R. Eliakim, M. Zaltshendler, and D. Zur, "Automated polyp detection system in colonoscopy using deep learning and image processing techniques." 2019.

[11] M. Min, S. Su, W. He, Y. Bi, Z. Ma, and Y. Liu, "Computer-aided diagnosis of colorectal polyps using linked color imaging colonoscopy to predict histology," *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[12] H. A. Qadir, Y. Shin, J. Solhusvik, J. Bergsland, L. Aabakken, and I. Balasingham, "Toward real-time polyp detection using fully cnns for 2d gaussian shapes prediction," *Medical Image Analysis*, vol. 68, p. 101897, 2021.

[13] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen *et al.*, "Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, vol. 7, no. 1, pp. 1–14, 2020.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[16] D. Hossain, M. H. Imtiaz, T. Ghosh, V. Bhaskar, and E. Sazonov, "Real-time food intake monitoring using wearable egocnetric camera," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 4191–4195.

[17] I. Perveen, M. M. Rahman, and M. Saha, "Upper gastrointestinal symptoms in general population of a district in bangladesh," *Journal of Enam Medical College*, vol. 4, no. 2, pp. 79–88, 2014.

[18] H. H. Publishing, "Understanding the results of your colonoscopy." [Online]. Available: https://www.health.harvard.edu/staying-healthy/understanding-the-results-of-your-colonoscopy

[19] "When possible, upper and lower gi endoscopies should be done on same day," May 2019. [Online]. Available: https://www.hopkinsmedicine.org/news/newsroom/news-releases/when-possible-upper-and-lower-gi-endoscopies-should-be-done-on-same-day

[20] N. N. Baxter, J. L. Warren, M. J. Barrett, T. A. Stukel, and V. P. Doria-Rose, "Association between colonoscopy and colorectal cancer mortality in a us cohort according to site of cancer and colonoscopist specialty," *Journal of Clinical Oncology*, vol. 30, no. 21, p. 2664, 2012.

[21] C. A. Doubeni, S. Weinmann, K. Adams, A. Kamineni, D. S. Buist, A. S. Ash, C. M. Rutter, V. P. Doria-Rose, D. A. Corley, R. T. Greenlee *et al.*, "Screening colonoscopy and risk for incident late-stage colorectal cancer diagnosis in average-risk adults: a nested case–control study," *Annals of internal medicine*, vol. 158, no. 5_Part_1, pp. 312–320, 2013.

[22] D. A. Corley, C. D. Jensen, A. R. Marks, W. K. Zhao, J. K. Lee, C. A. Doubeni, A. G. Zauber, J. de Boer, B. H. Fireman, J. E. Schottinger *et al.*, "Adenoma detection rate and risk of colorectal cancer and death," *New england journal of medicine*, vol. 370, no. 14, pp. 1298–1306, 2014.

[23] V. Conteduca, D. Sansonno, G. Ingravallo, S. Marangi, S. Russi, G. Lauletta, and F. Dammacco, "Barrett's esophagus and esophageal cancer: an overview," *International journal of oncology*, vol. 41, no. 2, pp. 414–424, 2012.

[24] J. C. Chang-Claude, J. Wahrendorf, Q. S. Liang, Y. G. Rei, N. Muñoz, M. Crespi, R. Raedsch, D. I. Thurnham, and P. Correa, "An epidemiological study of precursor lesions of esophageal cancer among young persons in a high-risk population in huixian, china," *Cancer research*, vol. 50, no. 8, pp. 2268–2274, 1990.

[25] M. Rahman, U. Dave, M. Hossain, M. Kibria, M. Nuruzzaman, F. Ahmed, A. Rowshon, M. Hasan *et al.*, "Pwe-142 state of the endoscopy services in bangladesh-first nationwide survey," 2016.

[26] H. R. Aslanian, F. K. Shieh, F. W. Chan, M. M. Ciarleglio, Y. Deng, J. N. Rogart, P. A. Jamidar, and U. D. Siddiqui, "Nurse observation during colonoscopy increases polyp detection: a randomized prospective study," *American Journal of Gastroenterology*, vol. 108, no. 2, pp. 166–172, 2013.

[27] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2018, pp. 122–129.

[28] T. Hirasawa, K. Aoyama, T. Tanimoto, S. Ishihara, S. Shichijo, T. Ozawa, T. Ohnishi, M. Fujishiro, K. Matsuo, J. Fujisaki *et al.*, "Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images," *Gastric Cancer*, vol. 21, no. 4, pp. 653–660, 2018.

[29] H. Nakahira, R. Ishihara, K. Aoyama, M. Kono, H. Fukuda, Y. Shimamoto, K. Nakagawa, M. Ohmori, T. Iwatsubo, H. Iwagami *et al.*, "Stratification of gastric cancer risk using a deep neural network," *JGH Open*, 2019.

[30] P. Mesejo, D. Pizarro, A. Abergel, O. Rouquette, S. Beorchia, L. Poincloux, and A. Bartoli, "Computer-aided classification of gastrointestinal lesions in regular colonoscopy," *IEEE transactions on medical imaging*, vol. 35, no. 9, pp. 2051–2063, 2016.

[31] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99–111, 2015.

[32] X. Zhang, F. Chen, T. Yu, J. An, Z. Huang, J. Liu, W. Hu, L. Wang, H. Duan, and J. Si, "Real-time gastric polyp detection using convolutional neural networks," *PloS one*, vol. 14, no. 3, p. e0214133, 2019.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[34] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[35] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 029–13 038.

[36] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding *et al.*, "Pp-yolo: An effective and efficient implementation of object detector," *arXiv preprint arXiv:2007.12099*, 2020.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[39] Y. Guo, Y. Li, L. Wang, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8368–8375.

[40] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn,"

in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[42] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[43] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[44] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[45] J. Xu, R. Zhao, Y. Yu, Q. Zhang, X. Bian, J. Wang, Z. Ge, and D. Qian, "Real-time automatic polyp detection in colonoscopy using feature enhancement module and spatiotemporal similarity correlation unit," *Biomedical Signal Processing and Control*, vol. 66, p. 102503, 2021.

[46] X. Liu, X. Guo, Y. Liu, and Y. Yuan, "Consolidated domain adaptive detection and localization framework for cross-device colonoscopic images," *Medical image analysis*, vol. 71, p. 102052, 2021.

[47] M. Laddha, S. Jindal, and J. Wojciechowski, "Gastric polyp detection using deep convolutional neural network," in *Proceedings of the 2019 4th International Conference on Biomedical Imaging, Signal Processing*, 2019, pp. 55–59.

[48] C. Cao, R. Wang, Y. Yu, H. Zhang, Y. Yu, and C. Sun, "Gastric polyp detection in gastroscopic images using deep neural network," *Plos one*, vol. 16, no. 4, p. e0250632, 2021.

Appendices

Appendix A: List of Acronyms

GI	**G**astro**i**ntestinal
CRC	**C**olo**r**ectal **C**ancer
ADR	**A**denoma **D**etection **R**ate
WCE	**W**ireless **C**apsule **E**ndoscopy
DMC	**D**haka **M**edical **C**ollege
CNN	**C**onvolutional **N**eural **N**etwork
SSD	**S**ingle **S**hot Multibox **D**etector
YOLO	**Y**ou **O**nly **L**ook **O**nce
TTA	**T**est **T**ime **A**ugmentation
P	**P**recision
R	**R**ecall
mAP	**M**ean **A**verage **P**recision
IoU	**I**ntersection **o**ver **U**nion
FPS	**F**rames **P**er **S**econd
NMS	**N**on **M**axima **S**uppression
SPP	**S**patial **P**yramid **P**ooling
CSP	**C**ross **S**tage **P**artial
RPN	**R**egion **P**roposal **N**etwork