Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering
Department of Electronics

Final Year Project Report Presented in Partial Fulfillment of the Requirements for the Degree of

MASTER

Title:

Gastrointestinal diseases diagnosis using capsule endoscopy and YOLOv8.

Presented by:

- **SAD SAOUD Abdel Djalil (Telecommunications)**
- **KORICHI Aymene (Computer Engineering)**

Supervised by:

- Prof. CHERIFI Dalila

Registration Number:...../2024

Abstract

Gastrointestinal (GI) diseases represent a substantial global health concern that annually affect millions of individuals and result in nearly two million deaths, highlighting the urgent need for early and accurate diagnosis, as undiagnosed cases can be life-threatening. Wireless Capsule Endoscopy (WCE) is a cutting-edge technology that enables the visualization of gastrointestinal diseases. By capturing thousands of frames per patient, it reduces the risk of human error and increases the accuracy of diagnoses. The enormous volume of images generated by WCE poses a significant challenge for manual diagnosis, prompting the development of computer-aided techniques to enhance the diagnostic process with high accuracy and within a short period.

Moreover, deep learning algorithms have demonstrated remarkable performance in medical imaging tasks and especially in GI diseases classification, leveraging the vast amounts of data generated by WCE to improve diagnostic precision and enhance patient outcomes.

This project represents a technique aimed at developing a robust YOLOv8-cls model for gastrointestinal disease classification. By training the model on the Kvasir dataset, the backbone of YOLOv8-cls learns to capture robust and informative features from the input images. These features serve as a powerful representation of the image content. Finally, the extracted features are fed into a classification head, which is fine-tuned to predict the class of the input WCE image, enabling accurate diagnosis of gastrointestinal diseases.

In our project we conducted a serie of four experiments to develop a high-performing YOLOv8-cls model for gastrointestinal disease classification. The initial experiment identified the best-performing YOLOv8-cls variant, which was then optimized using hyperparameter tuning. The final model was constructed using the mixture of experts technique, combining the best-performing variant with optimized hyperparameters. The top-performing variant from experiment 1 achieved an accuracy of 92.7%, but exhibited confusion between specific classes. Hyperparameter tuning and the mixture of experts approach improved the model's performance. Our proposed model achieved a testing accuracy of 96.25%, precision of 96.28%, and recall of 96.25%, with a 4% increase in testing accuracy, precision, and recall compared to the initial model.

Dedications

”In the name of Allah, the Most Merciful, the Most Compassionate. All praise be to Allah, the Lord of the worlds. May prayers and peace be upon the Prophet Muhammad, His servant and messenger.”

This study is wholeheartedly dedicated to our beloved parents, who have been a constant source of inspiration. They provided us with hope and encouragement during moments of doubt and despair, always guiding and supporting us. We are forever grateful to them.

Acknowledgments

First and foremost, we would like to extend our deepest gratitude to God Almighty, whose blessings and guidance have enabled us to successfully complete our project.

We are profoundly grateful to our supervisor, **Prof. CHERIFI Dalila**, for her invaluable advice, encouragement, and unwavering support throughout this endeavor. Her expert guidance and insightful feedback have been instrumental in shaping our research and future career prospects. We deeply appreciate her patience, dedication, and steadfast belief in our abilities.

We would also like to express our appreciation to **Prof. BAGHDADI Azeddine** for his significant contributions. His expertise and willingness to provide assistance have greatly enhanced the quality of this work.

Contents

List of Figures

List of Tables

List of Abbreviations

- **ANN** : Artificial Neural Networks.
- **ASHAScheduler** : Adaptive Successive Halving Scheduler.
- **CIoU** : Complete Intersection Over Union.
- **CNN** : Convolutional Neural Networks.
- **CPs** : Colorectal Polyps.
- **CRN** : Cancer Registry of Norway.
- **CSP** : Cross Stage Partial.
- **DL** : Deep Learning.
- **EC2** : Elastic Compute Cloud.
- **EGD** : Esophagogastroduodenoscopy.
- **EMA** : Exponential Moving Average.
- **EMR** : Endoscopic Mucosal Resection.
- **FLOPs** : Floating Point Operations per Second.
- **FN** : False Negative.
- **FP** : False Positive.
- **GBD** : The Global Burden of Diseases.
- **GEJ** : Gastroesophageal Junction.
- **GERD** : Gastroesophageal Reflux Disease.
- **GI** : Gastrointestinal.
- **GIou** : Generalized Intersection Over Union.
- **HPO** : Hyper-Parameter Optimization.
- **IRL** : Iterative Reinforced Learning.
- **LES** : Lower Esophageal Sphincter.

- **LMT** : Logistic Model Trees.
- **Lr0** : Initial Learning Rate.
- **ML** : Machine Learning.
- **MoE** : Mixture of Experts.
- **PANet** : Path Aggregation Network.
- **PPV** : Positive Predicted Value.
- **ReLU** : Rectified Linear Unit.
- **SAT** : Self-Adversarial Training.
- **SGD** : Stochastic Gradient Descent.
- **SIoU** : Signed Intersection Over Union.
- **SiLU** : Sigmoid-weighted Linear Unit.
- **SPP** : Spatial Pyramid Pooling.
- **SPPF** : Spatial Pyramid Pooling Fast.
- **TN** : True Negative.
- **TP** : True Positive.
- **UC** : Ulcerative Colitis.
- **VV** : Vestre Viken Health Trust.
- **WCE** : Wireless Capsule Endoscopy.
- **YOLO** : You Only Look Once.

General Introduction

Gastrointestinal diseases are one of the most prevalent and impactful disorders around the world. They can disrupt the function of the entire GI tract from the oesophagus to the rectum including stomach and intestines. These diseases can either acute; coming on suddenly and lasting only a short time, or chronic that persist for a extended time and require long-term management. They can develop because of functional or structural problems within the GI tract. Common factors contributing to these disorders include infections, autoimmune responses, lifestyle choices, and long-term use of certain medications. They can have a severe effect on patients' quality of life, causing them to experience debilitating symptoms like bloating, constipation, diarrhea, and exhaustion in addition to chronic pain and discomfort.

Diagnosing GI diseases involves a variety of methods to identify specific conditions accurately. The endoscopic examinations are the fundamental procedures that are used widely. The upper endoscopy examines esophagus, stomach and first part of small bowel, while the colonoscopy diagnoses large bowel and rectum. These two procedures can be also used for therapy, particularly in the removal of polyps after they are lifted and dyed to facilitate accurate identification of the polyp margins. Capsule endoscopy is another important technique, that entails swallowing a tiny, pill-sized camera that captures thousands of images as it passes through the digestive system. This method offers detailed views of regions that conventional endoscopies might miss, especially the small intestine.

Automatic detection of diseases using deep learning has shown significant promise in enhancing diagnostic accuracy and efficiency. This technology utilizes advanced architectures that can extract complex features and identify patterns and anomalies that may indicate the presence of disease. It can reduce the manual labor for medical personnel, lower average costs and significantly reduce mortality and the incidence of luminal GI diseases, improving overall patient's quality of life.

Our aim is to construct a robust deep learning model using YOLOv8 framework, capable of accurately predicting various pathological findings with minimal inference time. This model will leverage YOLOv8's impressive accuracy and speed to precisely identify abnormalities in GI tract images, ensuring reliable diagnosis that will improve the clinical efficiency.

This report consists of three chapters, the first chapter provides a review of the digestive system, describing its landmarks, the anatomy of both the oesophagus and large intestine and common diseases that can affect the GI tract. The second chapter is a theoretical background of our work that lists different related works, introduces the YOLO technology and its different versions, then describes the used model in detail in terms of its architecture and hyperparameters. The third chapter presents the different methods used in our task, the obtained results, and subsequent discussion for each experiment followed by a general discussion and a further work.

Chapter 1

Gastrointestinal diseases and methods of diagnosis

1.1 Introduction:

This chapter provides an general description of the digestive System, going through its main landmarks and the anatomy of both Esophagus and Large Intestine. It also delves into specific disorders such as esophagitis, ulcerative colitis, and polyps, discussing their causes, symptoms, diagnosis, and treatment. The anatomical and pathological features detailed in this chapter is crucial for accurately capturing the complexities of the digestive system, thereby improving the precision in distinguishing between normal conditions and potential disorders during the development of our model.

1.2 Digestive System Description:

The human digestive system consists of the gastrointestinal tract (GI tract), liver, pancreas, and gallbladder. The GI tract is a complex structure consisting of hollow organs interconnected in a lengthy and convoluted tube extending from the oral cavity to the anal opening. The hollow organs that make up the GI tract are the mouth, esophagus, stomach, small intestine, large intestine, and anus [1].

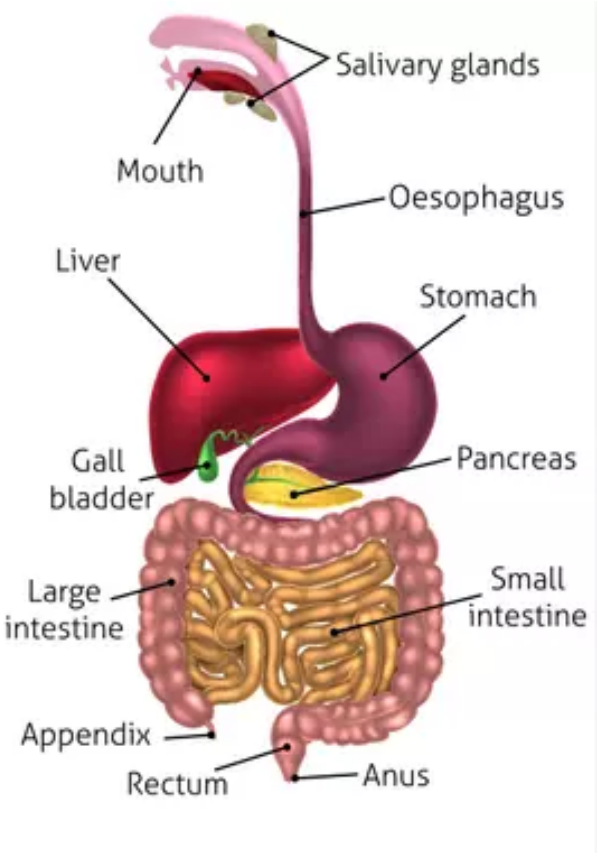

Figure 1.1: The digestive system [2].

The digestive system processes food through ingestion, mechanical and chemical digestion, absorption, and elimination. It breaks down food into nutrients, absorbs them into the bloodstream, and eliminates waste through defecation, providing the body with essential nutrients for energy and bodily functions.

1.3 Anatomical Landmarks:

An anatomical landmark refers to a distinctive feature discernible within the gastrointestinal (GI) tract, often readily observable through an endoscope. These landmarks play a crucial role in guiding endoscopic procedures and serve as reference points for pinpointing specific findings. Moreover, these landmarks frequently coincide with sites predisposed to pathology, such as ulcers or inflammation [3].

1. **Z-line:** The gastroesophageal junction (GEJ) holds significant clinical relevance owing to its pivotal role in various disease processes. This anatomical region is demarcated by the Z-line, serving as the transition site between the esophagus and the stomach. Under endoscopic examination, the Z-line presents as a distinct demarcation where the white mucosa in the esophagus meets the red gastric mucosa. Recognition and evaluation of the Z-line hold significance in diagnosing the presence of diseases like gastroesophageal reflux and describing pathology in the esophagus [3].

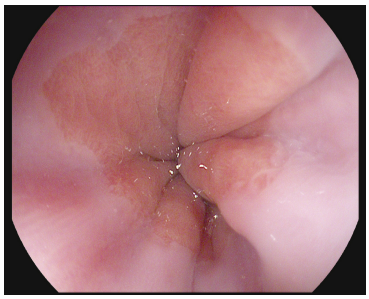

Figure 1.2: Normal Z-line [4].

2. **Pylorus:** The pylorus is a cone-shaped structure marking the division between the stomach and the small intestine. It functions to prevent the backflow of intestinal contents into the stomach and regulates the passage of food particles. Internally, it is lined with a mucous membrane that secretes gastric juices. The pyloric sphincter, controlled by circular muscle tissue, opens and closes to allow food passage. Typically, it remains open most of the time, permitting food to enter the small intestine [5].

Figure 1.3: Normal Pylorus [4].

3. **Cecum:** The cecum is the first part of the large intestine that digesting food enters after leaving the small intestine via the ileocecal valve, also called Bauhin's valve. Functioning as a liquid receptacle, it allows for the absorption of salts and

electrolytes from liquid waste, facilitated by muscular contractions. Reaching the cecum is the proof for a complete colonoscopy, which makes it important in the diagnosis [6].

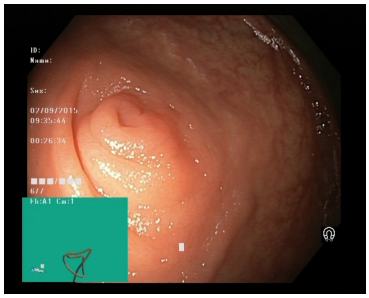

Figure 1.4: Normal Cecum [4].

1.4 Anatomy of the Esophagus:

The esophagus, a tubular organ of the digestive system, connects the pharynx to the stomach. Composed of mucosa-lined lumen, connective tissue, and smooth muscle, it measures about nine to ten inches in adults. The esophagus is subdivided into three anatomical segments: cervical, thoracic, and abdominal. With sphincters at each end, the upper esophageal sphincter permits food entry into the esophagus, while the lower esophageal sphincter facilitates passage into the stomach. The main function of the esophagus is to transport food entering the mouth through the throat and into the stomach [7].

1.5 Anatomy of the Large Intestine:

The large intestine is the last part of the gastrointestinal (GI) tract, which includes the cecum, appendix, entire colon, rectum, and anal canal. The colon can also be divided into segments: the ascending colon (traveling up), the transverse colon (traveling across to the left), the descending colon (traveling down), and the sigmoid colon (headed back across to the right). Unlike the small intestine, it has a shorter length but a much larger lumen. The colon mainly absorbs water and nutrients, compacts waste into feces, secretes potassium and chloride, and moves waste toward the rectum for elimination [8].

1.6 Gastrointestinal Diseases:

Gastrointestinal diseases affect the entirety of the gastrointestinal (GI) tract, extending from the oral cavity to the rectum. There are two types: functional and structural.

- Functional diseases are characterized by an abnormal movement of the GI tract while it looks normal when examined. Constipation, gas, bloating, and diarrhea are common examples.
- Structural diseases, on the other hand, affect the structure of the GI and also don't work properly. This includes diseases like hemorrhoids, diverticular disease, colon cancer, and inflammatory bowel disease [9].

Digestive disorders are widespread globally, causing significant distress and potential fatality. They often affect patients' quality of life and productivity. The Global Burden of Diseases, Injuries, and Risk Factors Study (GBD) 2019 shows that digestive diseases globally accounted for 2.56 million deaths and 2276.27 million prevalent cases [10]. In this work, we will focus on three main disorders: Esophagitis, Polyps, and Ulcerative Colitis.

1.6.1 Esophagitis:

Esophagitis is an inflammation or damage to the lining of the esophagus, which appears as a break in the esophageal mucosa. The severity of the inflammation is determined by the extent of these breaks in terms of length and the portion of the esophageal circumference affected. Professionals estimate that 1% of the population suffers from erosive esophagitis [11].

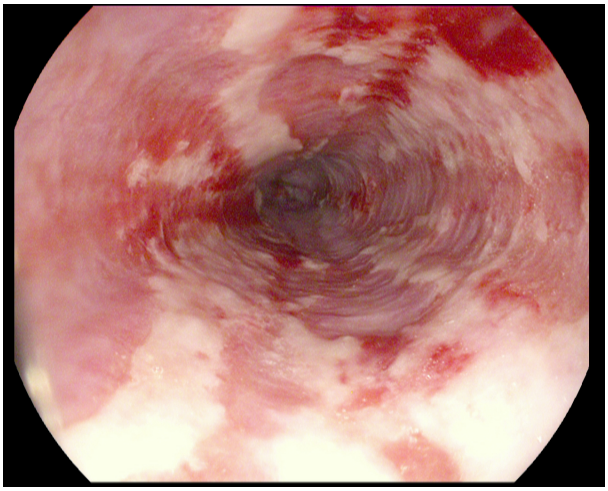

Figure 1.5: Esophagitis [4].

Inflammation of esophageal tissues can occur due to various factors such as immune responses to infections, allergic reactions, or corrosive injury. The following are the most widespread causes:

- **Reflux esophagitis:**
  Acid reflux, also known as gastroesophageal reflux disease (GERD), is a condition where stomach acid and occasionally stomach contents flow back up into the esophagus. Normally the lower esophageal sphincter prevents this backward flow of acid. However, if the LES relaxes abnormally or weakens, it can allow stomach acid to reflux into the esophagus. Repeated exposure to stomach acid can lead to irritation and inflammation of the esophageal lining [12].

- **Infectious esophagitis:**
  The Infection can result from various microorganisms, including bacteria, viruses, fungi, and parasites. Infectious esophagitis is rare and mostly occurs in people with poor immune system function [12].

- **Drug-induced esophagitis:**
  Oral medicines can cause tissue damage in the lining of the esophagus if they remain

for a long time in it.The most influntial drug include: Pain-relieving medications, Antibiotics and Quinidine [12].

- **Eosinophilic esophagitis:**
  Eosinophilic esophagitis likely occur if there is a high concentration of these white blood cells in the esophagus.this is mostly in response to an allergy-causing agent like food (milk, egg, wheat..) and pollen [12].

Common symptoms of esophagitis include a sore throat, heartburn, difficulty swallowing, and chest pain. The pain can range from mild to severe and may be constant or intermittent. Depending on the type of esophagitis, additional symptoms may include food getting stuck in the throat, nausea and vomiting, blood in the vomit, and mouth sores.

Diagnosing esophagitis generally begins with an evaluation of symptoms. A healthcare provider may initially prescribe acid-blocking medications; if these are effective, reflux esophagitis might be suspected. Further diagnostic procedures can include an upper endoscopy or an esophageal pH test for more detailed investigation [13].

A healthcare provider typically offers a combination of medication and lifestyle changes. Medications might include acid-blocking drugs to alleviate acid reflux symptoms. For conditions such as eosinophilic esophagitis, anti-inflammatory drugs and monoclonal antibodies may be prescribed. Lifestyle recommendations often focus on dietary changes to identify and avoid trigger foods, adjusting how medications are taken, and adopting an evening routine that includes consuming smaller meals well before bedtime to minimize acid reflux. Additionally, quitting smoking and reducing alcohol intake are advised to promote esophageal health[14].

1.6.2 Ulcerative Colitis:

Ulcerative colitis represents a pervasive challenge within the realm of gastroenterology, characterized by an idiopathic inflammatory process affecting the colon. It results in diffuse friability and superficial erosions on the colonic wall associated with bleeding.This inflammation restricted to the mucosa and submucosa of the colon. Based on where the inflammation is in your colon, healthcare providers classify UC into : Ulcerative proctitis(retum), Left-sided colitis(left side of the colon), Pancolitis(the entire colon). Ulcerative has no cure and affects both the physical and mental health. In the United States alone, accounts for a quarter-million provider visits annually, and medical costs are estimated to exceed four billion dollars annually[15].

The exact cause of ulcerative colitis is unknown, but several risk factors can contribute to its development, including [16]:

- **Genetics:** A first-degree relative of a patient with ulcerative colitis has a four times higher risk of developing the disease.
- **Race and Ethnicity:** There is a higher incidence in Jewish populations.
- **Gut Microbiome:** People with UC have differences in their microbiomes compared to those without the condition, indicating that alterations in the gut microbiota composition could lead to ulcerative colitis.

Regarding symptoms, patients usually notice signs of mild UC in its early stages, such as:

- Diarrhea Urgent bowel movements (sudden need to defecate).
- Tenesmus (feeling the need to defecate but being unable to).
- Mild abdominal (belly) cramping or tenderness.

The symptoms can become severe later [16], including:

- Blood, mucus, or pus in the stool.
- Severe abdominal cramping Fatigue (extreme tiredness).
- Sudden weight loss.
- Nausea.
- Fever.

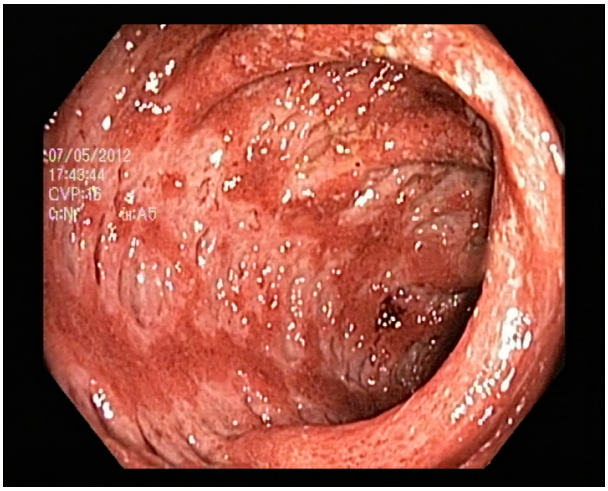

Figure 1.6: Ulcerative colitis [4].

Diagnosis involves various procedures. The healthcare provider may first perform a physical exam and inquire about symptoms and family history. They might order tests and procedures such as:

- **Blood Tests:** These can reveal indicators of anemia, suggesting possible bleeding in the colon or rectum.
- **Stool Samples:** These can show signs of infection, parasites, and inflammation in the stool.
- **Imaging Tests:**A barium enema can show a "stove-pipe" appearance in longstanding ulcerative colitis.
- **Endoscopic Tests:** Common endoscopic tests to diagnose UC include colonoscopy and sigmoidoscopy, which can reveal fragile and granular mucosa, loss of vascular pattern, and the presence of erosions and pseudopolyposis.

The selection of treatment depends on both the extent and severity of the disease. Healthcare providers may prescribe sulfasalazine and 5-aminosalicylates, with glucocorticoids added if needed. Thiopurines or biological drugs are considered if glucocorticoids fail. Regular colonoscopies are recommended for cancer risk. Colectomy, a surgical option is curative for ulcerative colitis complications[17].

1.6.3 Polyps:

The Researchers defined polyps as any mass protruding into the lumen of a hollow viscus. They are type of tumor or mass of abnormal cells that grow out of the mucous lining inside hollw organ like GI, nose or female reproductive organs. Colorectal polyps Can be either classified based on the macroscopic appearance into sessile (flat, arising directly from the mucosal layer) or pedunculated (extending from the mucosa through a fibrovascular stalk), or histologically classified as neoplastic or as non-neoplastic(hyperplastic, hamartomatous, or inflammatory). Neoplastic polyps are more important as they carry the risk of malignancy which is a significant stage in the progression towards colorectal cancer [18].

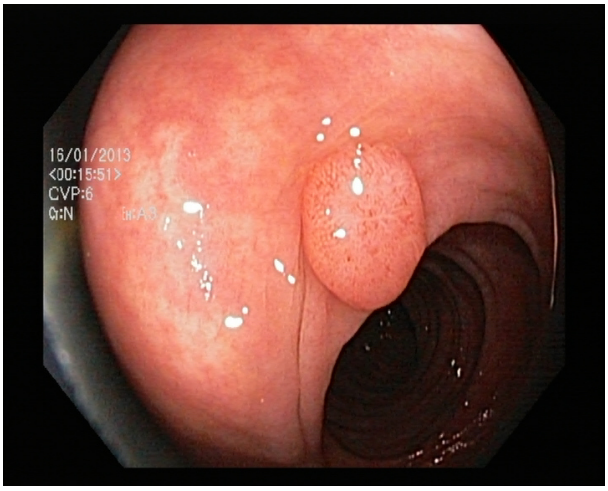

Figure 1.7: Polyps [4]

Most colon polyps occur randomly and don't have a obvious cause. However, research suggests that certain factors can raise the chance of develop it including [19]:

- Age
- Smoking and heavy alcohol use
- Family history
- Having inflammatory bowel disease
- Having obesity

Most colon polyps won't cause any symptoms, which makes screening important [19]. Possible symptoms may include bleeding from the rectum, blood in the stool, and feeling tired because of anemia.

Polyps Diagnosis can be detected with certain tests or procedures. Doctors usually ask for any symptoms or family history before performing a physical exam [20]. The procedures may include:

- Flexible sigmoidoscopy
- Colonoscopy
- Virtual colonoscopy

- Lower gastrointestinal series (barium enema)

The standard treatement is to remove the polyps using special tools during a colonoscopy or flexible sigmoidoscopy without surgery in most cases. After that doctors send it for testing to check for cancer. But once polyps are detected, doctors will likely recommend regular testing in the future due to an increased risk of developing additional polyps [21].

1.7 Polyp removal:

Large bowel polyps especially the neoplastic lesions are usually indicator of a cancer, making their removal during endoscopy a crucial procedure. Endoscopic mucosal resection (EMR) is a commonly used technique that involves injecting a liquid beneath the polyp to lift it off the underlying tissue. A snare is then used to catch and remove the polyp. The risk of mechanical or electrocautery damage to the GI wall's deeper layers is reduced by the lifting. Diluted indigo carmine dye is added to make the polyp margins easier to identify. Figure 1.8 illustrates dyed and lifted polyps. After the removal procedure, it is important to assess if the polyp has been removed entirely , as remaining polyp tissue may promote growth, and in the worst situation scenario, the emergence of malignancy. Figure 1.9 shows the resection site after a polyp is removed [22].

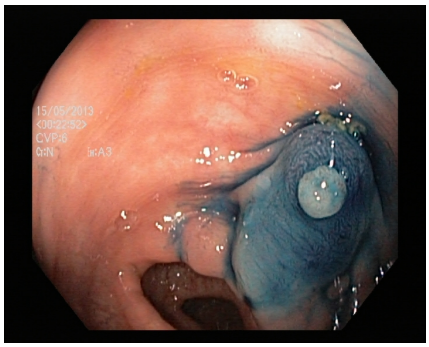

Figure 1.8: Dyed and Lifted Polyps [4].

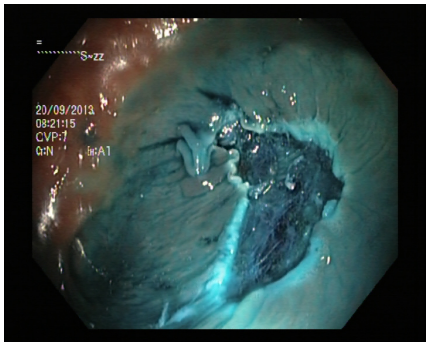

Figure 1.9: Dyed Resection Margins [4]

1.8 Endoscopic methods of diagnosis:

An endoscopy is a procedure done to examine the internal structures of the GI using a flexible viewing tube (endoscope). Most endoscopes have a light and special camera at the end which captures images or videos and displays them on a screen your provider sees. Common types of endoscopy include colonoscopy, upper endoscopy and capsule endoscopy.

1.8.1 Upper Endoscopy:

An upper endoscopy also known as esophagogastroduodenoscopy, or EGD is a procedure uses an endoscope to see the lining, or inside surface, of the upper GI tract. Upper GI endoscopy used to diagnose and treat symptoms related to the esophagus, stomach, and duodenum like heartburn, vomiting, removing polyps and treat bleeding. It is typically an outpatient procedure which takes between 10 to 20 minutes. Before the procedure, patience will likely be given a sedative to stay relaxed.

During the procedure, he will be asked to lie on his side, then the doctor will pass the tube carefully down his upper GI and pumps air through the endoscope to see the organs better. The dotor may take a sample of tissue, cells or fluids for testing, open up narrow sections, and other if needed [23].

1.8.2 Colonoscopy:

A colonoscopy is an examination of the inside of the large intestine, including colon, rectum and anus using endoscope. Colonoscopy can find causes for symptoms like bleeding from anus, diarrhea and abdomen pain, or screening colon polyps, colon and rectal cancer.

This procedure require special diet for a few days before (avoiding foods with a lot of fiber, clear liquid..), clean out the bowel using pill, liquid, or powder laxatives. It mostly like the upper endoscopy except that the doctor will insert the colonoscope through the anus[24].

1.8.3 Capsule Endoscopy:

Capsule endoscopy employs wireless technology to examine parts of the small intestine that can not be reached by other types of gastrointestinal (GI) endoscopy techniques.

The procedure entails swallowing a capsule, typically similar in size to a large vitamin tablet. This capsule contains a tiny camera, light, battery and radio transmitter that is encased in a tiny plastic shell.Figure 1.10 shows its different components.As the capsule moves through your digestive tract, the camera takes about 10,000 pictures and send them to a sensor device worn on the patient's abdomen so a healthcare provider can review them later. The capsule is passed naturally during bowel elimination.

Before the procedure, patients are required to provide the physician with a comprehensive medical history, including a list of current medications, and disclose the presence of a pacemaker or any other electromedical device. The physician asks about any issue that might affect the procedure like trouble swallowing and allergies. The patient have nothing to eat or drink for approximately 12 hours before the procedure.

Capsule endoscopy is known to be a safe, non-invasive test, but it is possible for the capsule to become trapped within the body which may require surgery to remove it [25].

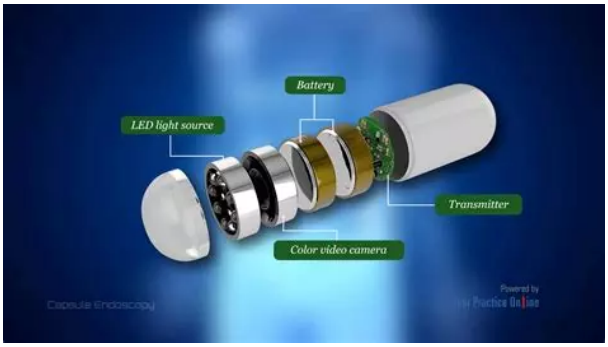

Figure 1.10: Capsule-Endoscopy components [26].

1.9 Summary:

In conclusion, Chapter 1 provided a comprehensive overview of the digestive system, exploring its main features and some common pathological conditions. Deep learning can provide automatic classification of endoscopic pathological findings in images captured within the GI tract through different approaches.

Chapter 2

Advanced Deep Learning Technique using YOLOv8 for Gastrointestinal diseases Classification

2.1 Introduction:

Deep learning (DL) is a rapidly advancing field within the medical domain, as it has the ability to enhance human lives by providing additional accuracy in diagnosis. Esteva et al [27] used 129,450 images of 2032 diseases and discovered that a DL network can diagnose the condition just as well as twenty-one dermatologists with board certification. Deep learning, branch of machine learning, uses multi-layered netwoks, called artifcial neural networks, or ANNs to simulate the information processing power of the human brain. Unlike ML methods that go through pre-processing, feature extraction, and careful feature selection , the learning of feature sets can be automated with DL using significant amount of data to map inputs to their labels[28].

Deep learning can be used in variety of applications including: image recognition and classification, natural language processing and finance. In order to build our classifier model, we used YOLO, which is one of the recent innovations in the realm of computer vision that has a huge impact in many fields. In this chapter will go through the detailed architecture of the used deep learning model, hyperparameters and optimization methods.

2.2 Related work:

The detection and the correct classification of gastrointestinal diseases at an early stage are significant challenges that need to be addressed for effective therapy and improved patient results. The field of medical imaging has been revolutionized by recent advancements in machine learning and deep learning, particularly in the detection and classification of gastrointestinal conditions. Researchers have proposed a variety of techniques that can be broadly categorized into two main approaches. The first approach utilizes machine learning, where hand-crafted features are extracted and then used to build the model. The second approach leverages deep learning, allowing the machine to autonomously extract features and construct the model[29]. Various studies have explored these advanced techniques to improve diagnostic capabilities in this domain.

Mosleh Hmoud Al-Adhaileh et al. [30] introduced three deep learning networks; GoogleNet, ResNet-50, and AlexNet, that were used for diagnosing lower gastrointestinal diseases using an enhanced version for only 5 classes of the Kvasir dataset by eliminating the classes of esophagitis, normal-z-line, and dyed lifted polyps due to the similarity of the images between classes that will lead to a confusion. Pretrained CNN models were fine-tuned using transfer learning and classified images into five classes. GoogleNet achieved an accuracy of 96.70% and a sensitivity of 96.60% with a validation accuracy of 96.70% within a training time of 608 min and 21 sec. ResNet-50 achieved a training accuracy and sensitivity of 96% and 94.80% respectively, achieving a validation accuracy of 97% within a training time of 693 min and 25 sec. AlexNet achieved superior results of 97% accuracy and 96.8% sensitivity within a training time of 100 min and 37 sec but a lower accuracy of 94.88% on validation data. A technique is proposed in [31] in which 17 methods were evaluated on a dataset of 4,000 images devided into 8 classes , splitted into training and testing sets, each containing an equal number of 2,000 images. The methods consist of machine-learning-based approaches, a 6-layer CNN, and transfer-learning approaches using Inception v3 and ResNet 50. The LMT classifier performed best among machine-learning approaches. The ResNet50 Features approach with the LMT classifier had the best overall performance, achieving an accuracy of 95.8%, a precision of 82.9%, and a recall of 82.6%. Misclassifications were most common with

esophagitis and Z-line classes. A similar dataset of 4000 images was used by Andrea Asperti et al. [32], where an ensemble of models approach was employed, leveraging transfer learning from pre-trained convolutional neural network models. To enhance the model's robustness and generalization, they used Keras' utilities for data augmentation, applying random transformations to training instances such as zooming out and normalization while generating new images during training. The model achieved a performance of 91.5% in accuracy, precision, and recall. The study Xu Zhang et al. [33] introduced the Gastric Precancerous Disease Network (GPDNet), a model for classifying three types of gastric precancerous conditions. The GPDNet uses fire modules from SqueezeNet to decrease its size and speed up classification. To maintain high accuracy with fewer parameters, an innovative method, Iterative Reinforced Learning (IRL), is employed. IRL fine-tunes near-zero parameters after initial training, and the adjusted model is used for further training. The application of IRL improves accuracy by about 9% after six iterations leading to a final accuracy of 88.90%. An advanced neural network architecture has been employed in [34] to automatically identify ulcers and erosions in pictures captured by Wireless Capsule Endoscopy (WCE). This methodology, used image cropping and compression. The AlexNet convolutional neural network was trained on the preprocessed dataset to effectively differentiate between abnormal lesions and healthy tissue.accuracy rates of 95.16% and 95.34% respectively. Additionally, the sensitivity rates were 96.80% for ulcers and 93.67% for erosions, while the specificity rates were 94.79% for ulcers and 95.98% for erosions. The study [35] employed a Single Shot MultiBox Detector, which is a deep convolutional neural network (CNN). The Convolutional Neural Network (CNN) was trained using a dataset of 16,418 photos, consisting of 4,752 colorectal polyps (CPs) and 4,013 normal colorectum images. The performance of the CNN was then evaluated using a separate set of 7,077 colonoscopy images, which included 1,172 CP images from 309 distinct CP types. A total of 1246 CPs were discovered, with a sensitivity rate of 92% and a positive predictive value (PPV) of 86%. Out of the polyps that were identified, 83% were correctly categorized.

2.3 YOLO Overview:

YOLO (You Only Look Once) is a state-of-the-art, real-time object detection system. Unlike earlier methods, which either use sliding windows or detect the objects using two stages, where the first stage identifies potential regions containing objects or proposals while the second phase employs a classifier to these proposals, YOLO use a single pass of the network and regression to predict the detection outputs making it much faster,more accurate, and better at generalization [36]. YOLO models have been employed in many fields like autonomous vehicle systems, agriculture, cancer detection, security systems and traffic applications. Over the last decade, YOLO family has persistently evolved at a rapid pace. The YOLO versions are discribed bellow:

1. **YOLOv1:** YOLOv1 was developed by Ross Girshick, Ali Farhadi, Santosh Divvala, and Joseph Redmon. It divides the image into a S $\times$ S grid,each grid predicts multiple bounding boxes and confidence, filters-out low-confidence predictions, and remove overlapping boxes.
   The network architecture is based on GoogLeNet; consists of 24 convolutional layers followed by two fully connected layers [37].

2. **YOLOv2:** Built upon the error analysis of its predecessor, this version shows a significant localization errors and low recall. The representation data was made easier to learn by adding batch normalization in all its convolutional layers, high resolution classifier to spot smaller objects, and anchor boxes to predict shape and size of objects more accurately[38].

3. **YOLOv3:** Introducing a new backbone architecture called Darknet-53, this version combines the features used in YOLOv2 and Darknet-19 with residual connections connect the input of the 1×1 convolutions with the output of the 3×3 convolutions. YOLOv3 also predicts objects at three different scale to detect objects of broader range of sizes [39].

   At this period, the overall architecture of object detectors are divided into three parts:backbone, neck and head. Figure 2.1 shows the high-level structure.
   The backbone extracts features at varying scales, the neck collects the backbones' features for further transformations, and the head is responsible for making predictions.

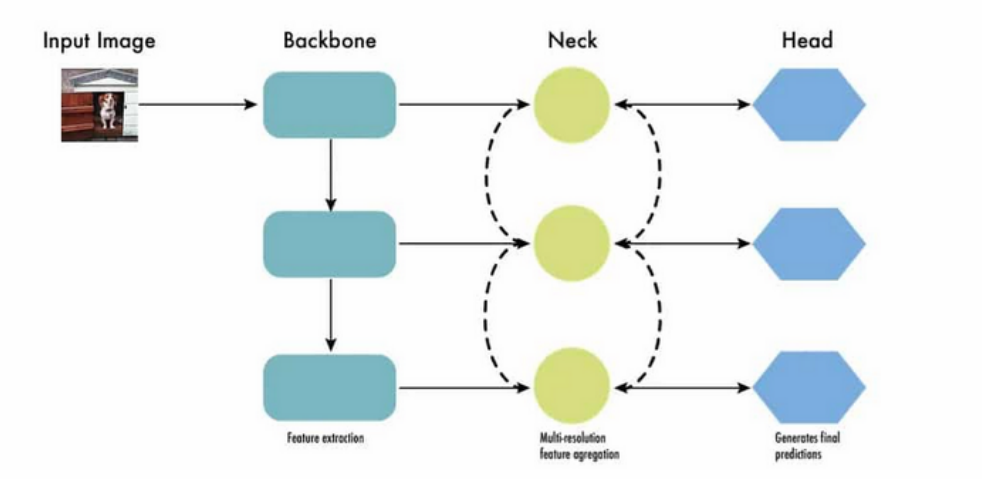

Figure 2.1: The high-level architecture of object detectors [40]

4. **YOLOv4:** YOLOv4 improvements were based on two approaches: "bag-of-freebies" which increase training cost and "bag-of-specials" which increase the inference cost. For the first strategy, YOLOv4 introduce Mosaic, and Self-Adversarial Training (SAT) data augmentation methods, and use DropBlock, CIoU loss and Cross mini-bath normalization. For the second strategy, it employed CSPDarknet53 backbone, SPP additional module, PANet path-aggregation neck, and YOLOv3 head. YOLOv4 also used a genetic algorithms to get the optimal hyperparameters[41].

5. **YOLOv5:** Developed by Ultralytics using PyTorch, this model follows the same architecture as the previous version with two primary changes, SPP structure is replaced with SPPF which more than doubles the speed of processing, and 3x3 structure with a 6x6 Conv2d structure.
   YOLOv5 applies a variety of training strategies to improve the performance like: multiscale training,AutoAnchor and Exponential Moving Average (EMA).It is can be used for object detection, classification and segmentation tasks[42].

6. **YOLOv6:** This version used a new backbone based on RepVGG called EfficientRep , Rep-PAN neck and efficient decoupled head.YOLOv6 also came up with a new classification VariFocal loss and SIoU/GIoU regression loss.For label assignment ,it uses Task alignment learning which was first proposed in TOOD [43].

7. **YOLOv7:** This architecture featured three main improvements:an extended efficient layer aggregation network which controls the shortest longest gradient path and combines the features of different groups to improve the learning process, an enhanced concatenation-based models which maintain the structure of the model, and the "bag-of-freebies" approach includes re-parametrization for better performance[36].

8. **YOLOv8:** YOLOv8 is a new computer vision model built by Ultralytics that can be used for object detection, image classification, pose estimation and instance segmentation tasks. The figure 2.2 shows a diagram of the network's architecture. YOLOv8 is anchor free that predict the center directly without using an offset from a known anchor box resulting in a faster detections. Also, the first 6x6 convolutions was replaced by 3x3, and C2f block replaced C3. YOLOv8 employs different data augmentation techniques; at each epoch, the model sees a different variation of the images, but then it turn it off for the last ten training epochs[44].

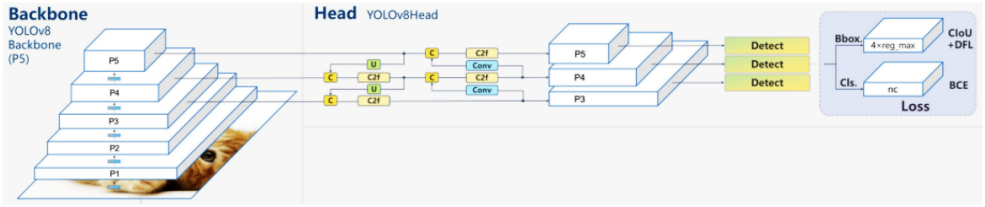

Figure 2.2: YOLOv8 Architecture diagram [45]

2.4 Image Classification using YOLOv8 for GI diseases diagnosis:

This model is primary designed for object detection tasks, but it supports many computer vision tasks including image classification. We will use the classify models as we aim to indetify the disease regardless of its location in the image.

Image classification involves assigning a confidence score and a single label to an entire image from a set of predefined categories based on extracted features. However, this process encounters various challenges that affect its accuracy and efficiency including: variability in image quality, background clutter,data imbalance and high computational resource requirements. To address these issues, classification models may require advanced data augmentation techniques, transfer learning and robust algorithm development. The architecture of the chosen model uses a variant of the EfficientNet to perform classification. EfficientNet Developed by Google researchers was based on the compound scaling concept which introduces a compound coefficient as a user-defined parameter that uniformly scales the neural network's dimensions [46]. The YOLOv8 architecture consists of a backbone and a head. The backbone is responsible for feature extraction whereas

head processes these features to make final predictions. Figure 2.3 illustrates the overall structure of this architecture.

Figure 2.3: The overall architecture of YOLOv8 classification model.

2.4.1 Backbone:

The backbone is a pre-trained network used to extracts features hierarchically; the first layers capture low-level features (e.g., edges, textures) while deeper layers get higher-level features (e.g., shapes, entire objects). This network helps reducing the spatial resolution of the image and increasing its channels. The figure bellow illustrates the overall architecture of the backbone

YOLOv8 backbone is essentially similar to that of YOLOv5. It consists of a sequence of two main blocks: Conv and C2f blocks; where the input image is 640x640x3 by default. Each Conv layer reduces the spatial dimensions by half due to the stride of 2, while the number of kernels typically increases by a power of 2 at each stage starting from 64 kernels. In contrast,the C2f blocks apply multiple convolutional operations with the same number of kernels obtained from the previous Conv layer, preserving the spatial dimensions while refining the features. The number of output channels for each block change according to the size of the model (see subsection 2.5.1).

2.4.1.1 Conv blocks:

The convolutional (conv) blocks in YOLOv8 are essential for detecting features in images using learnable filters. The figure 2.4 shows its layers.

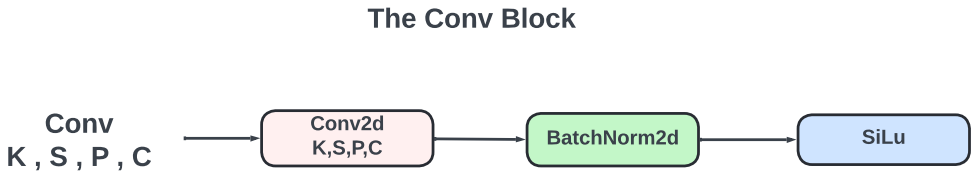

Figure 2.4: The convolutional Block.

The following are the typical components of the Conv layer:

- **The Conv2d layer:** is the core building block of any CNN network. The Conv2d consists of learnable kernels. Althought they are usually small in spatial dimensionality, but they extend throughout the full depth of the volume. The convolutional layer outputs a 2D activation map by convolving each filter across the input's spatial dimensionality. The process illustrated in Figure 2.5.

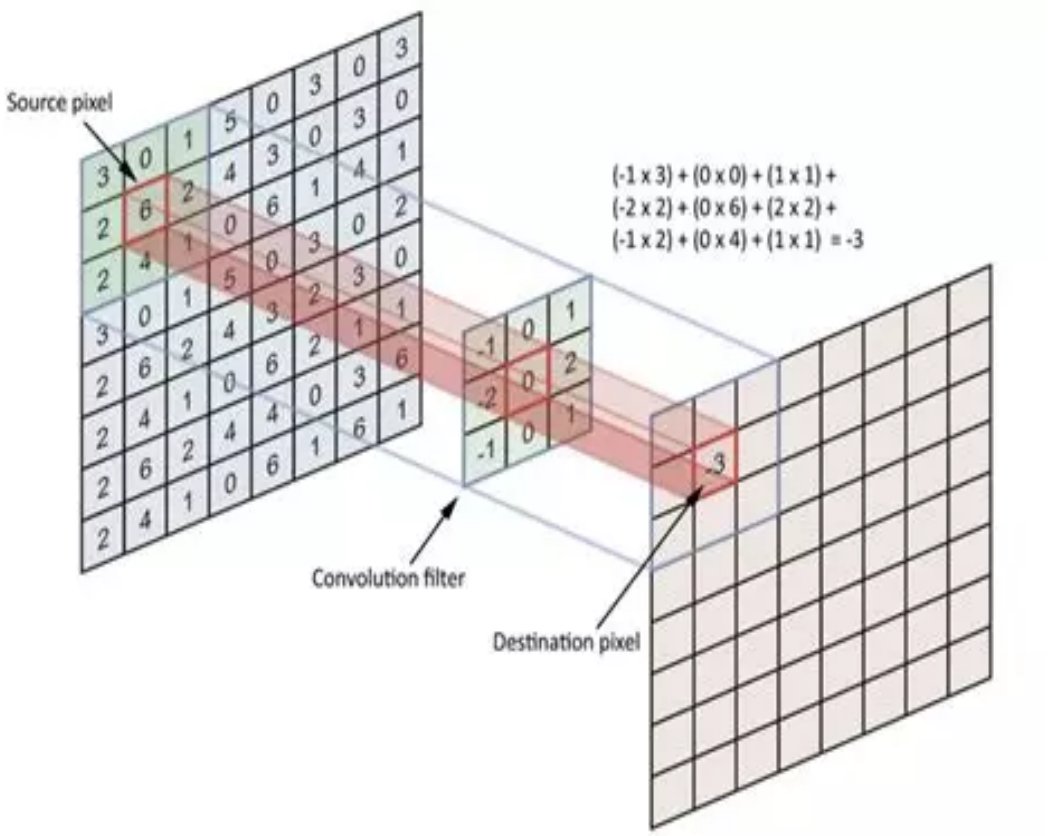

Figure 2.5: The convolution operation [47].

This layer has three hyperparameters, the depth, the stride and the padding. The stride refers to the step size with which the convolutional filter moves across the feature map, wheras the padding the process of adding extra pixels around the input to give further control of output's dimensions which be can calculated with the following equation:

$$\frac{(V - R) + 2.Z}{S + 1} \tag{2.1}$$

Where V represents the input volume size R represents the receptive field size, Z is the amount of zero padding applied, and S refers to the stride[48].

- **BatchNorm2d:** The Batch Normalization Layer, introduced by Sergey Ioffe and Christian Szegedy in their 2015 paper, has been widely used in many architectures. This technique aims to stabilize and accelerate the learning process by normalizing each batch through the layers. For the inference, it uses the accumulated moving averages of mean and variance. The proposed algorithm shown in figure 2.6.

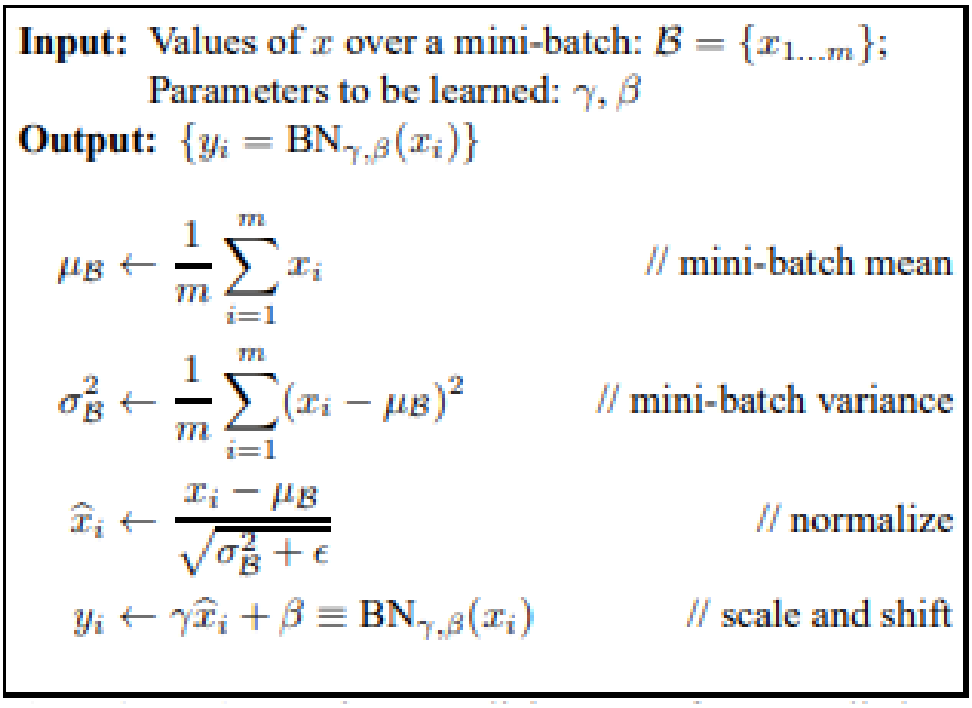

Figure 2.6: The algorithm of Batch Normalizing Transform [49].

By maintaining a stable distribution of activations, normalization prevents internal Covariate Shift due to the updating of network parameters during training, and vanishing/exploding gradients [49].

- **SiLU:** SiLU or Sigmoid-weighted Linear Unit, is an activation function proposed for reinforcement learning application. This function calculated using the following equation [50]:

$$a_k(z_k) = z_k\sigma(z_k) \tag{2.2}$$

Where the sigmoid function is [50]:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

The SiLU and ReLU graphs can be visualized, as seen in Figure 2.7. The two activation functions are approximately equal for z_k values of large magnitude. Unlike common activation units, SiLU is a non-monotonic function with a global minimum value of approximately -0.28 for $z_k \approx -1.28$. This characteristic serves as a soft floor, contributing to learning stabilization and weight regularization [50].

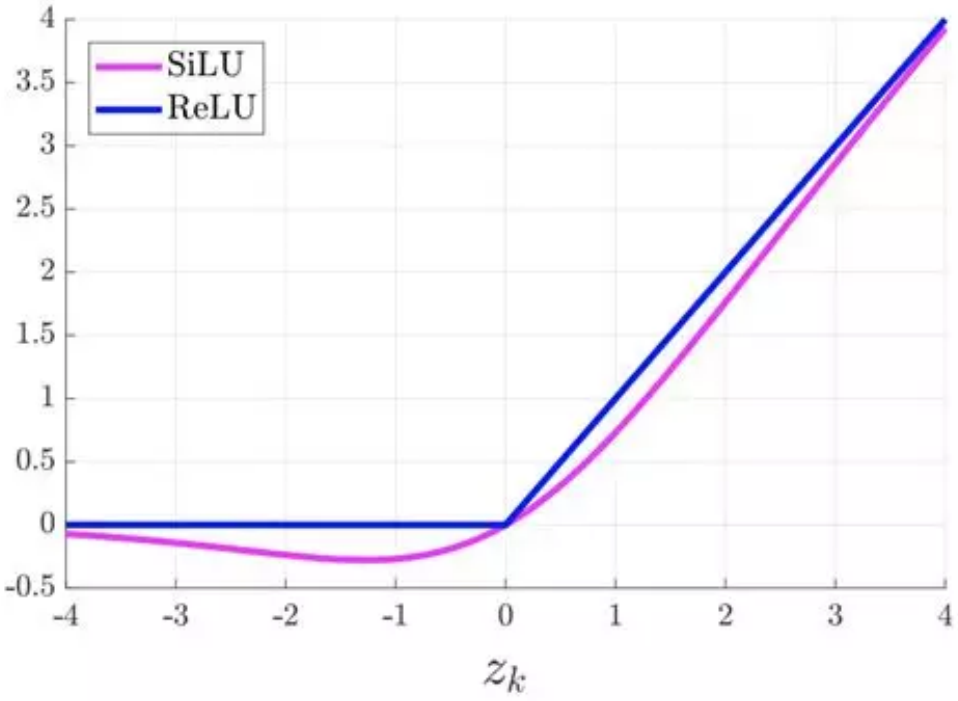

Figure 2.7: The activation functions of the SiLU and the ReLU [50].

2.4.1.2 C2f:

C2f is new block introduced in the YOLOv8 architecture to replace C3 layer of YOLOv5 [51]. The figure 2.10 below shows the difference between them. This module combines high-level semantic features with low-level spatial information by concatenating the outputs of all the Bottleneck layers.

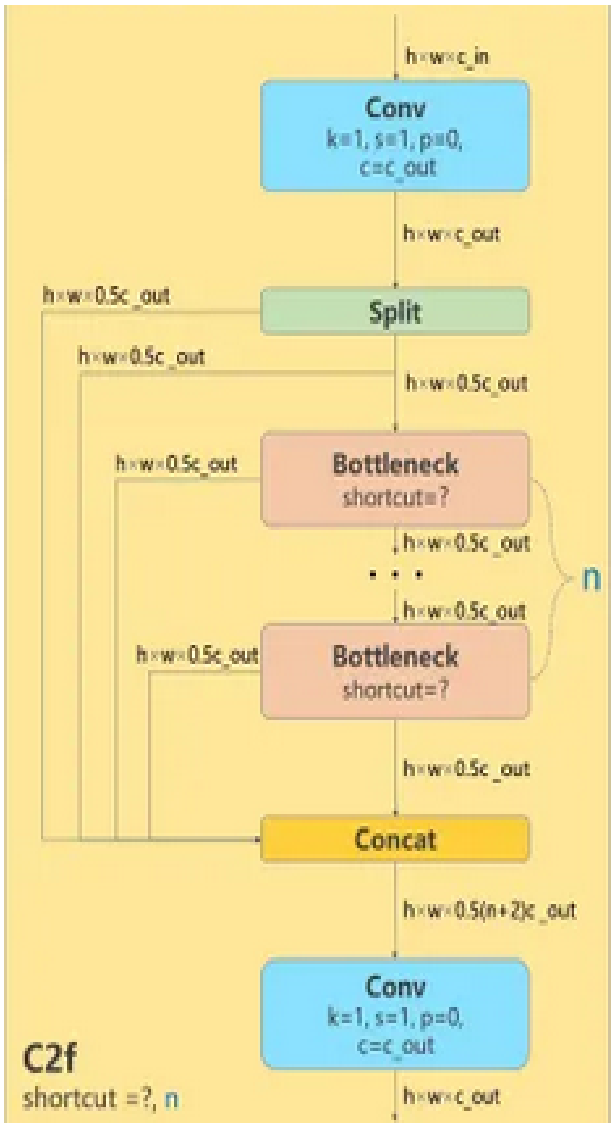

Figure 2.8: C2f

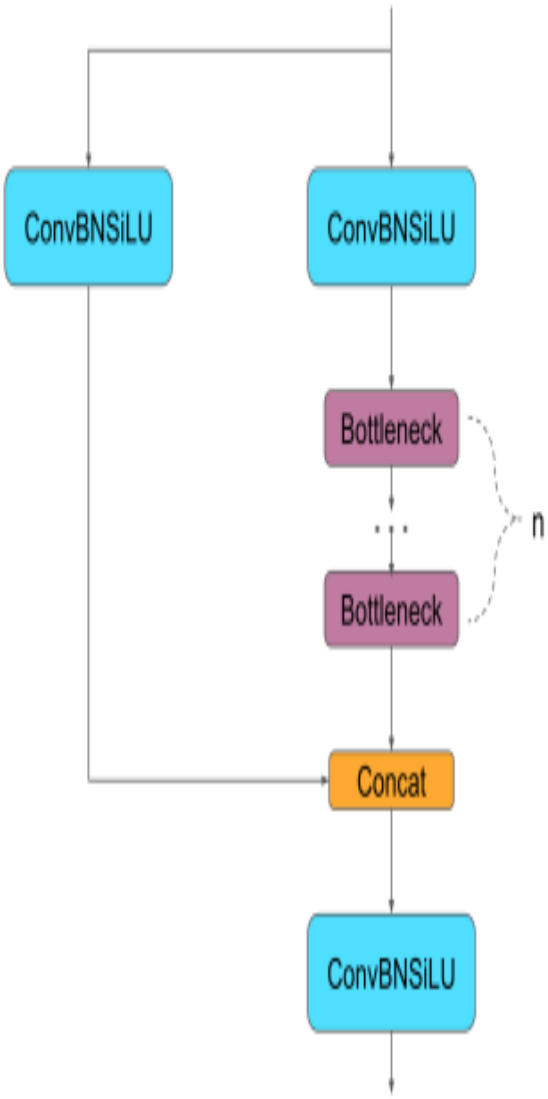

Figure 2.9: C3

Figure 2.10: The difference between C2f and C3 blocks [45].

The main used layers used in the C2f are listed below:

- **Conv Block:** Initial convolutional layer with kernel size 1, stride 1, padding 0, and output channels equal to c_out.
- **Split:** Splits the input feature map into two halves, each with dimensions (h, w, 0.5c_out).
- **Bottleneck Layers:**The bottleneck uses two 3x3 Conv blocks of stride of 1 with an optional shortcut, similar to the ResNet block described in 2015 [52], which is described in the figure 2.11 bottleneck block.

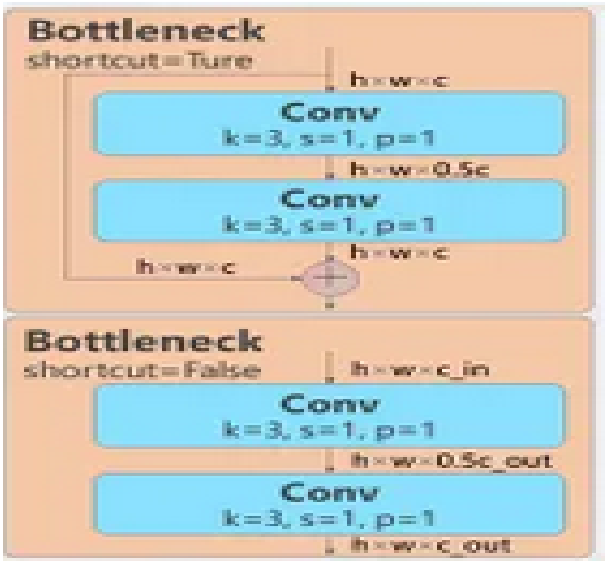

Figure 2.11: The bottleneck block [45].

- **Concat:** Concatenates the outputs of all Bottleneck layers to combine high-level semantic features and low-level spatial information.
- **Final Conv Block:** a final layer with kernel size 1, stride 1, padding 0, and output channels equal to c_out.

2.4.2 Head:

The head is the final block of the model that makes the final predictions based on the features extracted by previous layers. Mainly, the head maps the high-dimensional features that the model has learned into an output space that is appropriate for particular tasks like detection, regression, or classification. A typical head classifier uses fully connected layers, regularization techniques and activation functions. The YOLOv8 classification models' head consist of the following layers:

- **Convolutional layer:** The head make further feature extraction using 1x1 Conv layer with stride of 1. This layer reduces the number of channels while preserving the spatial dimensions, allows the model to learn a weighted combination of the channels, and introduces non-linearity with the SiLU activation.
- **Adaptive average pooling layer:** is a commanly used layer in CNNs that aims to outputs a fixed-size feature map regardless of the size of the input. Unlike normal pooling layers that uses a fixed kernel size, the adaptive average pooling dynamically changes the window size to get the target output dimension, and calculates the average value of the input map within each kernel [53]. The feature map will be downsampled to a single value per channel. The figure 2.12 illustrates this opeartion.

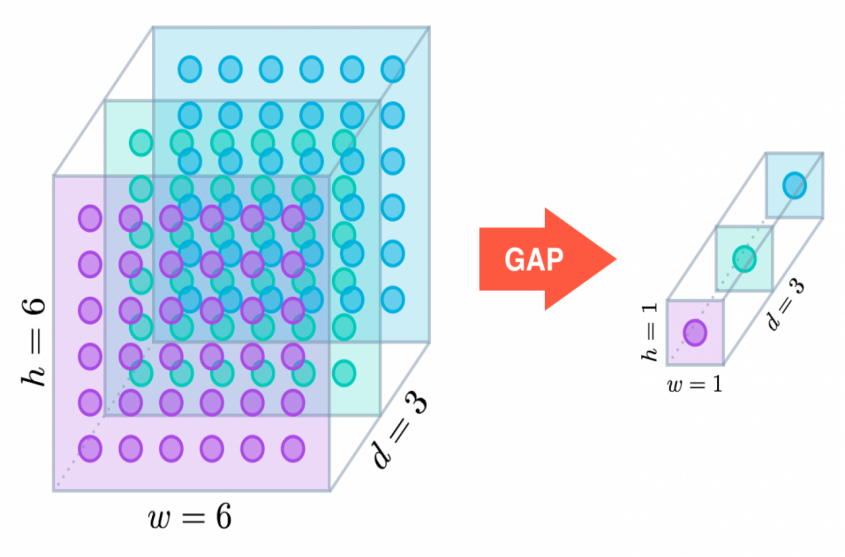

Figure 2.12: The average pooling operation [54].

- **Dropout:** is a regularization technique used in neural networks to prevent overfitting where randomly selected neurons are ignored during training. Dropout introduces random noise to allow neurons to learn general features rather than memorizing the training data. For inference, the dropout is disabled to use the full capacity of the model, and the activations also are scaled by the dropout rate to have the same expected values. YOLOv8 set dropout to 0 by default, but it can be configured according to their specific needs [55].

- **Fully connected layer:** also known as a dense layer, is type of layer in which each neuron is connected to every neuron in the previous layer. Each input scaled and shifted with a learnable weight and bias and went thought an activation function. The output of a fully connected layer is given by the equation [56]:

$$y_{jk}(\mathbf{x}) = f\left(\sum_{i=1}^{n_H} w_{jk}x_i + w_{j0}\right) \tag{2.4}$$

  Where $y_{jk}(\mathbf{x})$ is the output of the j-th neuron in the k-th layer for input $\mathbf{x}$. The function f is the activation function. The term $\sum_{i=1}^{n_H} w_{jk}x_i$ is the weighted sum of inputs x_i with weights w_{jk} and bias w_{j0}.

- **Softmax:** is an activation function used for multi-class classification tasks. It transformers the raw data(also called logits) from the preceding layer into a probability distribution over multiple classes which can be used to predict the class with the highest probability. The softmax is defined as [57]:

$$\sigma(\mathbf{z}) = \frac{e^{\mathbf{z}}}{\sum_{j=1}^{N} e^{z_j}} \tag{2.5}$$

Where $\sigma(\mathbf{z})$ is the output vector of probabilitie, $\mathbf{z}$ is the input vector of logits. The Cross-Entropy Loss is commonly used with Softmax in classification tasks using this equation [58]:

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \tag{2.6}$$

Where y_i is the true label, $\hat{y}_i$ is the predicted probability, and output size is the number of outputs.

2.5 YOLOv8 variants for classification:

In our work,we examine five different classify models: Nano (YOLOv8n), Small (YOLOv8s), Medium (YOLOv8m), Large (YOLOv8l), and Extra Large (YOLOv8x). YOLOv8 Nano is the fastest and smallest model, whereas YOLOv8 Extra Large (YOLOv8x) is the most precise but slowest among the options, allowing for rapid deployment and task-specific customizatio. These models are pretrained on the ImageNet dataset, which has 1000 different classes. This section delves into the architectural differences and performance metrics of these models.

2.5.1 Architectural Differences:

The YOLOv8's complexity is defined by three coefficients which are: depth multiplier (d), width multiplier (w) and max channels (mc). These coefficients directly affect the the number of bottlenecks in the C2f block and the number of output channels for each block. The table 2.1 below shows the different parameters for each model.

The number of bottlenecks in the C2f block is calculated with this formula $n = m \times d$, where $m = 3$ for the first and last C2f blocks and $m = 6$ for the intermediate blocks. Both the width multiplier and max channels parameter control the number of output channels according to this equation:

$$\min(\text{num\_kernel}, \text{mc}) \times w \tag{2.7}$$

Where num_kernel is the number of kernels for each block [59].

Model Variant	**Depth Multiplier (d)**	**Width Multiplier (w)**	**Max Channels (mc)**
YOLOv8n	0.33	0.25	1024
YOLOv8s	0.33	0.50	1024
YOLOv8m	0.67	0.75	1024
YOLOv8l	1.00	1.00	1024
YOLOv8x	1.00	1.25	1024

Table 2.1: Architectural Differences of YOLOv8 Models [59].

2.5.2 Performance Metrics:

The resulting performance metrics were gotten using the ImageNet validation set with an image size of 224X224. The models were performing differntly in terms of of accuracy, inference speed. These performance metrics are detailed in the table 2.2 [60].

- **Accuracy:** is measured in terms of top-1 and top-5 accuracy. YOLOv8n achieves 69.0% (Top-1) and 88.3% (Top-5), whereas YOLOv8x reaches up to 79.0% (Top-1) and 94.6% (Top-5).
- **Inference Speed:** The speed metrics were averaged over validation images using an Amazon EC2 P4d instance, utilizing both CPU ONNX and A100 TensorRT. YOLOv8 Nano has an inference speed of 12.9 ms on CPU and 0.31 ms on A100 TensorRT. In contrast, the YOLOv8 Extra Large (YOLOv8x) model has an inference speed of 232.0 ms on CPU and 1.01 ms on A100 TensorRT.
- **Parameters:** The number of parameters also scales with the model size. YOLOv8n has 2.7 million parameters, while YOLOv8x has 57.4 million parameters.
- **FLOPs:** The computational complexity measured in FLOPs (Floating Point Operations per Second) increases from 4.3 billion in YOLOv8n to 154.8 billion in YOLOv8x.

Model	Top-1 Acc.	Top-5 Acc.	Speed (CPU ONNX)	Speed (A100 TensorRT)	Params (M)	FLOPs (B)
YOLOv8n-cls	69.0%	88.3%	12.9 ms	0.31 ms	2.7	4.3
YOLOv8s-cls	73.8%	91.7%	23.4 ms	0.35 ms	6.4	13.5
YOLOv8m-cls	76.8%	93.5%	85.4 ms	0.62 ms	17.0	42.7
YOLOv8l-cls	78.3%	94.2%	163.0 ms	0.87 ms	37.5	99.7
YOLOv8x-cls	79.0%	94.6%	232.0 ms	1.01 ms	57.4	154.8

Table 2.2: Performance metrics of YOLOv8 classification models[60].

2.6 Hyperparameters:

The hyperparameters are top-level parameters that are explicitly set by the user before the training process. Unlike the learnable parameters of the models, hyperparameters remains constant during the training. They are used to improve the model's performance in terms off speed, accuracy and generalization[61]. Through the development process, we adjusted several hyperparameters to get the optimal performance. The common used ones are listed below:

- **Batch size:** represents the number of samples that are processed before updating the internal model parameters. Batch size can range from one sample to the entire training data. Choosing the optimal batch size in machine learning involves a trade-off. Small batches require more updates and may have noisy gradients, but they also lead to better generalization. Large batches, on the other hand, speed up epoch completion, but they may not generalize well and require more memory.

- **Epochs:** refers to how many times the whole dataset is fed forward and backward through the neural network during training. Training for multiple epochs allows the model to learn complex pattern leading to better performance. However, too many epochs may cause overfitting.

- **Learning rate:** dictates magnitude of the steps the model takes during training while moving toward a minimum of a loss function. The selection of a learning rate is crucial; a high rate allows the model to learn more quickly by making bigger steps, but it also increases the risk of overshooting the minimum loss. On the other hand, small rate can lead to slow convergence, and get the model stuck in a local minima. Various strategies can be used to manage the learning rate values. Learning rate schedules, such as step decay, exponential decay, and piecewise constant decay,change the learning rate within predetermined intervals or according to a mathematical formula. Adaptive learning rate algorithms like AdaGrad, RMSprop, and Adam update the learning rate based on the gradients or parameters. YOLOv8 uses the learning rate warm-up approach; the warmup_epochs hyperparameter define the number of epochs in which the learning rate gradually increasing the from a low value to the initial learning rate to stabilize training early on [59].

- **Momentum:** is a factor used in SGD and Adam optimizers that determines the extent to which past gradients affect the current update. It helps in reducing noisy gradients, and accelerating the convergence.

- **Weight decay:** is a scaling factor in the regularization term of the Weight decay technique. This technique used to prevent overfitting by adding a penalty term that is proportional to the sum of the squared weights. The L2 regularization updates the parameters according to this equation [62]:

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha \nabla f_i(\theta_t), \tag{2.8}$$

  where θ_t is the parameter vector at iteration t, λ is the regularization parameter, α is the learning rate, and $\nabla f_i(\theta_t)$ is the gradient of the loss function f_i with respect to θ_t.

- **Dropout:** determines the fraction of neurons to be randomly dropped out during training in a neural network. It takes values between 0 to 1. This technique aims to prevent overfitting and learn more robust and generalized features.

- **Optimizer:** aims to minimize a predefined loss function and find the optimal weights for the network. Various algorithms employ distinct strategies to improve the performance, speed, and stability of the model. YOLOv8's optimizer can be either defined before training or dynamically selected based on model configuration. In our work we used the following optimizers:

  - **SGD:** or Stochastic Gradient Descent is a popular optimization technique that updates the model parameters by calculating the gradient of the objective function with respect to one example at a time, rather than using the entire dataset using the following formula[63]:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}, y^{(i)}) \tag{2.9}$$

Where, θ is the parameter vector, η is the learning rate, $\nabla_\theta J(\theta; x^{(i)}, y^{(i)})$ is the gradient of the cost function J with respect to θ for the i-th training example, $x^{(i)}$ is the input, and $y^{(i)}$ is the output. This approach allows for faster, more frequent updates and requires less memory, making it suitable for large datasets, but it may also lead to instability in convergence.

– **RMSprop:** is a gradient-based algorithm proposed by Geoff Hinton that normalizes the gradient using a moving average of squared gradients before updating the parameters.It based on these two rules [63]:

$$E[g_t^2]_t = 0.9E[g_t^2]_{t-1} + 0.1g_t^2 \tag{2.10}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g_t^2]_t + \epsilon}} g_t \tag{2.11}$$

Where $E[g_t^2]_t$ represents the exponentially weighted average of the squared gradients at time step t, which helps to smooth out the gradient updates. The term g_t is the gradient of the cost function at time step t, while θ_t denotes the parameter vector at time step t. The learning rate is given by η, and ϵ is a small constant added to prevent division by zero. Hinton suggests to set η to 0.001 and γ to 0.9 as default value.

This will decrease the step if gradients are large avoiding exploding and increase the step for small gradients to avoid vanishing .

– **Adam:** is an adaptive learning rate algorithm that combines the advantages of two methods:AdaGrad and RMSProp. This technique offers faster convergence with little memory requirements. The figure below shows the proposed algorithm. It updates both the exponential moving averages of the gradient and the squared gradient, and corrects their initial bias towards zero [64].

– **Adamw:** is an improved version of Adam, since Wilson and al show that Adam and similar adaptive gradients techniques do not generalize like SGD with momentum. This method decouples the weight decay from the gradient-based update to make it only proportional to the weight itself [62]. The difference between Adam and Adamw was illustrated in figure 2.13.

Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

given $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$
initialize time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second moment vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$
repeat
  $t \leftarrow t + 1$
  $\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow$ SelectBatch$(\boldsymbol{\theta}_{t-1})$ ▷ select batch and return the corresponding gradient
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) + \lambda \boldsymbol{\theta}_{t-1}$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$ ▷ here and below all operations are element-wise
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
  $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t / (1 - \beta_1^t)$ ▷ β_1 is taken to the power of t
  $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t / (1 - \beta_2^t)$ ▷ β_2 is taken to the power of t
  $\eta_t \leftarrow$ SetScheduleMultiplier(t) ▷ can be fixed, decay, or also be used for warm restarts
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left(\alpha \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) + \lambda \boldsymbol{\theta}_{t-1} \right)$
until *stopping criterion is met*
return optimized parameters $\boldsymbol{\theta}_t$

Figure 2.13: The difference between Adam and Adamw optimizers [62].

This optimizer yields better training loss, and generalize better than models trained with Adam.

2.7 Hyperparameters optimization:

Hyperparameter optimization (HPO) is an iterative process aimed at finding the right combination of hyperparameter values to get the best model's performance metrics. The used algorithms differ in how they handle the exploration vs. exploitation trade-off. The most used methods are given as follows:

- **Grid search:** also known as parameter sweep, is an algorithm that evaluates every combination of values in the specified subset of the hyperparameters space. It usually uses cross-validation to evaluate the model reliably. Although it is simple to implement, grid search is computationally expensive, and suffers from curse of dimensionality.
- **Random search:** also known as stochastic search,is an optimization method used to fine-tune models by randomly sampling values from a predefined hyperparameter's distribution. Random search is faster and more efficient than grid search for high-dimensional search spaces, but it might miss the optimal hyperparameter values. The figure 2.14 below shows the differnce between the grid and random search methods.

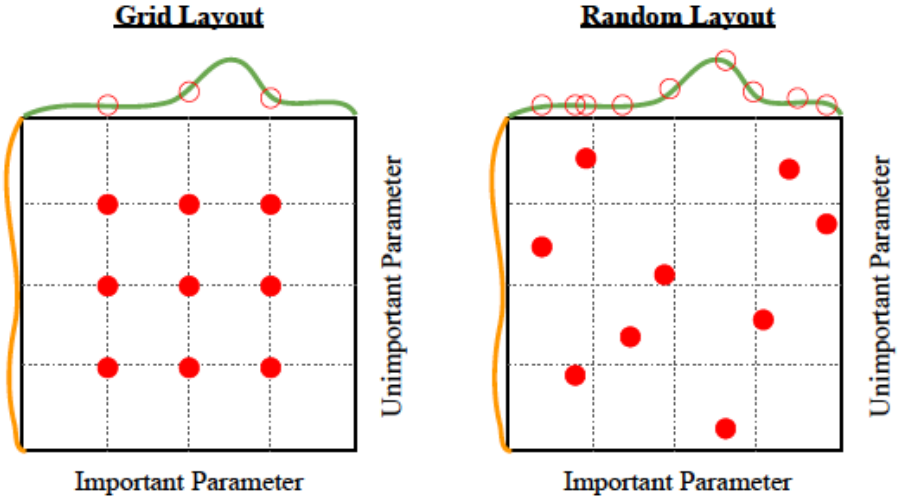

Figure 2.14: The difference between random and grid search [65].

- **Bayesian search:** is an efficient optimization algorithm that based on bayesian statistics. It consider the previous hyperparameter values and their performance while determining the next set of hyperparameters to evaluate. Initially, it samples random hyperparameter values and evaluate them on the actual model. The obtained results are used to train a surrogate model, such as a Gaussian process or random forest, which can predict performance and uncertainty for new hyperparameters. The acquisition function then selects the best candidates by balancing exploration and exploitation, guiding the search towards potentially promising areas. The chosen hyperparameters will be evaluated on the original model to update

the surrogate model with the resulting performance. The process continues until a stopping criterion like fixed number of iterations or convergence to optimal hyperparameters is met. The figure 2.15 shows the different elements of this method. This approach shows faster convergence as it leverages past evaluations to inform future searches [66].

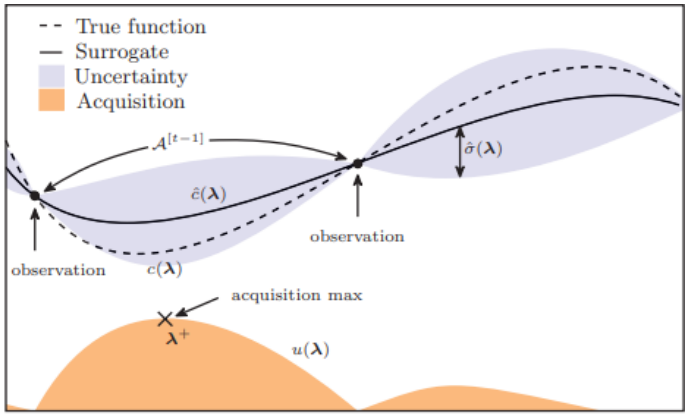

Figure 2.15: The Bayesian search algorithm [66].

2.8 Model ensembling:

Model ensembling is a powerful machine learning paradigm in which a number of models often referred to as "weak learners" are trained to solve a given problem and then combined to produce better performance and balance the bias-variance trade-off.there three main algorithms that used to combine the weak learners [67]:

- **Bagging:** This method aggregates multiple models that trained on different samples of the dataset created through bootstrapping.
- **Boosting:** Boosting uses sequentially training models, where each model attempt to correct the error of its predecessors.The learners usually combined with a weighted sum.
- **Stacking:** This algorithm is a two-layered approach, that learns several different weak learners in the first stage then uses a meta-model to combine the predictions to make the final decision.

The proposed model for our work uses the mixture of experts approach, which is a variant of the stacking method. This technique distribute the dataset to different models and uses a trained gate to select the appropriate model or combination of experts to each task. The figure 2.16 illustrates a general structure of this approach.

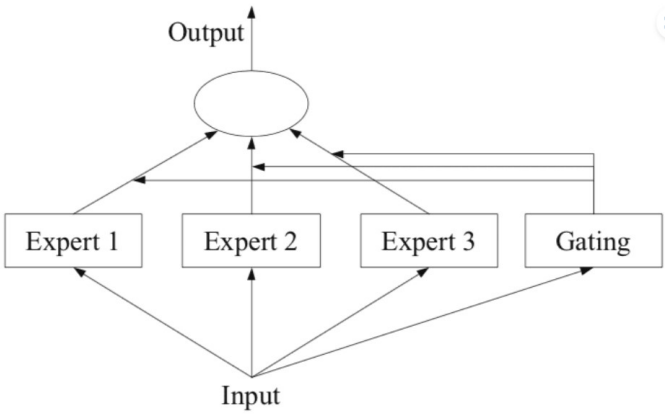

Figure 2.16: Mixture of expert network with 3 experts [68].

2.9 Summary:

In this chapter, we provided a comprehensive overview of the various YOLO object detection models, focusing on the YOLOv8 variants' architecture of classification, their hyperparameters and the optimization algorithms. These models will be utilized for the automatic classification of endoscopic pathological findings in images captured from the gastrointestinal (GI) tract, demonstrating their potential to enhance diagnostic accuracy and efficiency in medical image analysis.

Chapter 3

Experiments & Results

3.1 Introduction:

This chapter presents a comprehensive evaluation of the YOLOv8 model for the classification of gastrointestinal diseases using the Kvasir-v2 dataset of Wireless Capsule Endoscopy (WCE) images. Leveraging the YOLOv8 architecture, pretrained on large-scale image datasets, we aim to investigate its efficacy in accurately identifying and classifying abnormalities within endoscopic imagery.

3.2 Tools:

In this project, we strategically combined tools to boost the efficiency of our classification tasks. Python was the foundational language for code development. We also used The Ultralytics library to easily load, train YOLOv8 models and automatically generate and visualize the results, while Ray Tune was utilized in optimizing YOLOv8's performance by fine-tuning its hyperparameters for improved accuracy, enabling the exploration for optimal model configurations in our classification endeavors.

1. **Python:** Python is the most popular, interpreted, open-source and a high-level programming language ; known for its simplicity, readability, and versatility. The language's popularity can be attributed to its ease of use, accessibility, and its extensive ecosystem of libraries and frameworks; particularly in the field of machine learning, which make it suitable for data scientists and researchers [69].

2. **Ultralytics:** The Ultralytics library represents a robust tool for computer vision applications, offering an extensive selection of deep learning models and algorithms suitable for tasks like object detection, segmentation, and classification [70]. Ultralytics' seamless integration with popular deep learning frameworks enables researchers and developers to address complex visual recognition challenges with ease, driving advancements in various industries, including autonomous driving and medical imaging. Its extensive documentation and active community make it a valuable resource for individuals seeking to use deep learning for practical computer vision applications.

3. **Ray Tune:** Ray Tune is an advanced hyperparameter tuning library aimed at streamlining and enhancing the process of discovering optimal configurations for machine learning models. Developed on top of the Ray distributed computing framework, Ray Tune offers a comprehensive toolkit catering to users of all levels, facilitating efficient exploration and adjustment of hyperparameters across a diverse array of machine learning algorithms and frameworks [71].

3.3 Dataset:

3.3.1 Dataset Description:

The Kvasir dataset was collected using endoscopic equipment at the Bærum Hospital's gastroenterology department, and precisely annotated by medical experts from Vestre Viken Health Trust (VV) and the Cancer Registry of Norway (CRN) [72].

The Kvasir-v2 dataset consists of 8,000 images categorized into 8 different classes; each

containing 1,000 images that feature anatomical landmarks like the Z-line, pylorus and cecum, alongside pathological findings such as esophagitis, polyps, and ulcerative colitis. It also includes images related to lesion removal procedures like "dyed and lifted polyp" and "dyed resection margins", with resolutions ranging from 720x576 to 1920x1072 pixels. Additionally, some of the images contains a small green picture in picture that illustrates the position and configuration of the endoscope inside the bowel, captured using an electromagnetic imaging system (ScopeGuide, Olympus Europe). This supplementary information may be valuable for subsequent investigations, but should be handled carefully when detecting the endoscopic findings [73].

This dataset Provides a robust foundation for tasks like image retrieval, machine learning and deep learning, thereby enhancing research in computer aided gastrointestinal disease classification and cancer prevention.

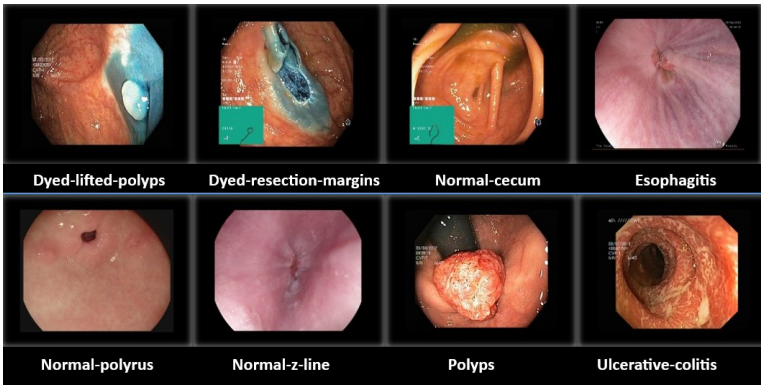

Figure 3.1: Samples from the Kvasir dataset.

3.3.2 Dataset splitting for YOLOv8 Image Classification:

To effectively utilize our dataset for image classification with the YOLOv8 model, it is essential to divide the dataset into separate subsets for training, validation, and testing. Partitioning the dataset into training, validation and testing subsets ensures that the model is trained, validated, and tested on separate. This approach provides an accurate evaluation of the model's performance and its ability to generalize to unseen data. Our dataset is divided as follows:

- **Training subset:** Comprising 70% of the total images. This subset is used to train the YOLOv8 model, allowing it to learn the intricate patterns and features within the data.
- **Validation subset:** Consisting of 20% of the total images used for the validation of our model.
- **Testing subset:** Consisting of 10% of the total images. This subset is reserved for evaluating the final performance of the model on unseen data.

3.3.3 Data Augmentation and Preprocessing:

In our project, YOLOv8-cls incorporates augmentation and preprocessing techniques into the training process by default, eliminating the need for external augmentation and

preprocessing. These techniques play a vital role in the training pipeline, enhancing the model's ability to generalize and adapt to variations in input data. The specific augmentation and preprocessing arguments utilized by YOLOv8-cls are as follows [74]:

- **Hsv-h:** Adjusts the hue of the image, introducing color variability and aiding in generalization across diverse lighting conditions. The default value is 0.015, falling within the range of 0.0 to 1.0.
- **Hsv-s:** Alters the saturation of the image, impacting color intensity and facilitating the simulation of different environmental conditions. The default value is 0.7, within a range of [0.0-1.0].
- **Hsv-v:** Modifies the brightness of the image, assisting the model in performing well under varying lighting conditions. The default value is set to 40%.
- **Translate:** Shifts the image horizontally and vertically, aiding in the detection of partially visible objects. The default value is 0.1.
- **Scale:** Resizes the image by a gain factor, simulating objects at different distances from the camera. The default value is set to 0.5.
- **Fliplr:** Flips the image horizontally with a specified probability, beneficial for learning symmetrical objects and enhancing dataset diversity. The default flipping value is 0.5.
- **Mosaic:** Combines four training images into one, replicating diverse scene compositions and object interactions. This technique is particularly effective for complex scene understanding. The default value is set to 1.0 (100%).
- **Erasing:** Randomly erases a portion of the image during classification training, encouraging the model to focus on less obvious features for recognition. The default probability is 0.5, within the range [0-0.9].
- **Crop-fraction:** Crops the classification image to a fraction of its size, emphasizing central features and adapting to object scales to reduce background distractions. The default value is 1.0, within the range [0.1-1.0].

By incorporating these techniques, the model learns to adapt to different scenarios, improving its performance and resilience to variations in lighting conditions, object sizes, and spatial configurations.

3.4 Evaluation Metrics:

In the realm of evaluation metrics, particularly in the context of multi-class classification, it is essential to define key components of the confusion matrix to facilitate a comprehensive analysis [75].

3.4.1 Confusion Matrix components:

1. **True Positive (TP):** Represents instances where the model correctly predicts the positive class, indicating accurate identification of samples where both the prediction and actual classification are positive. Specifically, it pertains to the number of diseased images which are accurately classified [76].

2. **True Negative (TN):** Signifies accurately identified negative instances, where the model correctly predicts the negative class, indicating that both the prediction and the actual outcome are negative. In the context of endoscopic image analysis, TN refers to the number of normal images which are correctly classified [76].

3. **False Positive (FP):** Denotes erroneously identified samples, often referred to as "false alarms," where the model incorrectly predicts the positive class, encompassing diseased images which are incorrectly classified [76].

4. **False Negative (FN):** Reflects incorrectly identified negative instances, where the model wrongly predicts the positive class. In the context of endoscopic image analysis, FN refers to The number of normal images which are incorrectly classified [76].

By understanding and utilizing these components effectively, a thorough evaluation of the model's classification performance in multi-class scenarios can be achieved, providing valuable insights for advanced analysis and decision-making in research endeavors at the master's level.

3.4.2 Accuracy:

Accuracy (ACC) is an important measure that shows the percentage of correctly identified samples out of all predicted samples [77]. It's calculated by dividing the total of True Positives (TP) and True Negatives (TN) by the total number of samples; as shown in the following equation [77]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

3.4.3 Loss:

Loss functions are crucial in machine learning as they measure the difference between predicted outputs and actual target values [78]. This disparity assessment is vital for evaluating model performance and guiding the learning process to improve predictions. By identifying the discrepancy between predictions and actual values, loss functions facilitate the optimization of machine learning models and enhance their predictive accuracy.

3.4.4 Precision:

Assesses the model's accuracy in classifying samples as positive and prevent to misclassify a negative sample as Positive [79] It is calculated by dividing the number of correctly classified positive samples by the total number of samples labeled as positive, including both correct and incorrect classifications. Precision is determined using the

following formula [79]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

3.4.5 Recall:

Recall assesses the model's capacity to identify positive instances without caring if all the negative samples were incorrectly classified as Positive. A higher recall indicates that the model is more effective in detecting positive samples [79].The recall is calculated as ratio of true positives (TP) to the sum of true positives and false negatives (FN). Mathematically, it is represented as [79]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

3.5 Experimental procedure for GI diseases classification using YOLOv8-cls:

In our project, we conducted a series of four experiments to evaluate the YOLOv8 classification models on the Kvasir dataset. The first experiment involved evaluating all YOLOv8-cls model variants (n, s, m, l, x) on the original 8-class Kvasir dataset to determine the model with the best performance in terms of training accuracy, loss, and testing accuracy, while also considering the training time. The second experiment involved training a the best performing model from experiment 1 on a Kvasir dataset with 5 classes to evaluate its performance in comparison to related works. Following this, In the third experiment, we applied hyperparameter tuning using Ray Tune on the best-performing YOLOv8-cls model identified in the first experiment. This systematic optimization process allowed us to determine the optimal hyperparameters configuration for training, further enhancing the model's performance. Lastly, a mixture of experts approach was implemented to construct our final model and enhance its performance using the best hyperparameters determined in the previous experiments.

3.6 Experiments:

In this section, we present a detailed account of the experiments conducted and the corresponding resultant findings, emphasizing the performance of the YOLOv8-cls models in the classification of gastrointestinal diseases.

3.6.1 Experiment 1: Comparative Study of YOLOv8-cls Variants on the Full Kvasir Dataset

The first experiment aimed to assess the performance of YOLOv8-cls on the complete 8,000-images Kvasir dataset across 8 classes. Concurrently, a comparative study will be conducted on all YOLOv8-cls variants (n, s, m, l, x) to determine the best-performing model based on metrics such as loss, accuracy, precision, recall and time. This comprehensive analysis will guide the selection of the most effective YOLOv8-cls variant for

accurate and efficient image classification tasks. Figure 3.2 represents a general structure for this experiment.

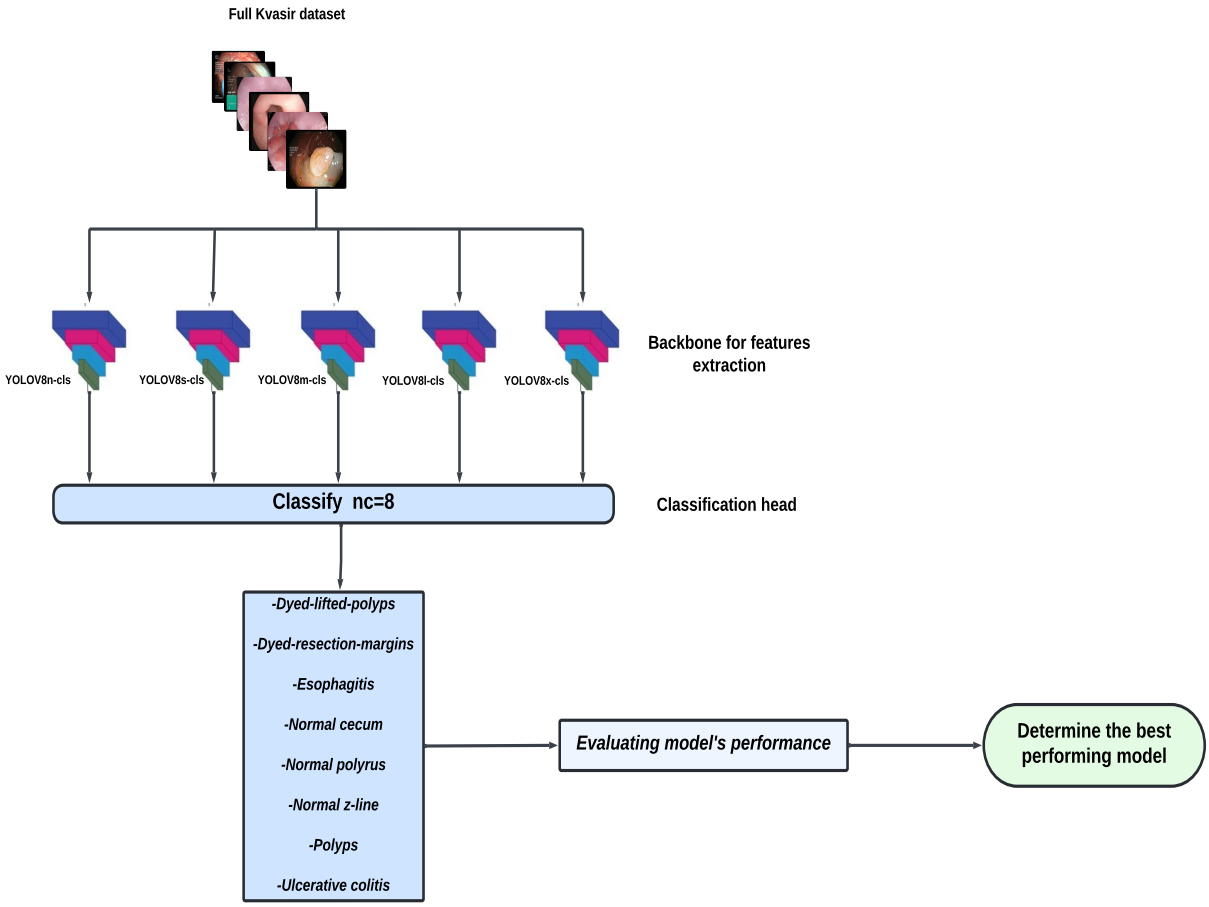

Figure 3.2: Structure of evaluating YOLOv8-cls variants.

We aimed to train five YOLOv8-cls variants (n, s, m, l, x) on the full Kvasir dataset and using the default hyperparameters, augmentation, and preprocessing values set by YOLOv8 to select the best-performing model in terms of loss, accuracy, performance on unseen data and execution time. We utilized the default hyperparameters configuration, with a batch size set to 16 and the optimizer set to auto, allowing the system to select an appropriate optimizer, initial learning rate, and momentum. Specifically, the chosen hyperparameters for the optimizer were AdamW (lr0=0.000714, momentum=0.9). The image size was maintained at 640. Each of the five YOLOv8-cls model variants (n, s, m, l, x) was trained on the dataset for 15 epochs using a Tesla T4 GPU.

Results: Throughout the training process, we closely tracked key metrics such as training loss, validation loss and top-1 accuracy to ensure the models were learning effectively and converging to optimal performance. The obtained results are summarized in the table 3.1, which presents the top-1 accuracy, training loss, validation loss and training time for each YOLOv8-cls variant (n, s, m, l, x).

Model	Accuracy_top1	Training loss	Validation loss	Training time (hours)
YOLOv8n-cls	92.298 %	0.22737	1.3825	0.922
YOLOv8s-cls	93.049 %	0.1984	1.3599	0.998
YOLOv8m-cls	93.488 %	0.10147	1.3533	1.036
YOLOv8l-cls	94.177 %	0.08609	1.3485	1.157
YOLOv8x-cls	93.8 %	0.0937	1.348	1.414

Table 3.1: Performance Metrics of YOLOv8-cls Models on Kvasir Dataset.

Upon completion of training, we evaluated the performance of each model on the testing data to assess their accuracy, precision and recall on unseen data. The results are illustrated in the following table 3.2 :

Model	Testing accuracy	Precision	Recall
YOLOv8n-cls	91.69 %	91.66 %	91.68 %
YOLOv8s-cls	92.2 %	92.27 %	92.24 %
YOLOv8m-cls	92.7 %	92.61 %	92.58 %
YOLOv8l-cls	92.2 %	92.07 %	92.02 %
YOLOv8x-cls	92.2 %	92.28 %	92.18 %

Table 3.2: Performance Metrics of YOLOv8-cls Models on testing data.

Discussion: The experimental results highlight several key findings. While the YOLOv8l-cls model achieved the highest top-1 accuracy of 94.177% and the lowest training loss of 0.08609 in 1.157 hours of training, YOLOv8x-cls achieved a top-1 accuracy of 93.8% with the longest training time of 1.414 hours. Its performance on testing data aligned with that of YOLOv8l-cls and YOLOv8s-cls at a testing accuracy of 92.2%, a precision of 92.28% and a recall of 92.18%. In contrast, YOLOv8n-cls, the quickest with an training time of 0.922 hours, attained the lowest top-1 accuracy, the lowest performance on testing data and exhibited the highest losses, suggesting it is less effective for this dataset.
YOLOv8m-cls achieved a top-1 accuracy of 93.488% and a competitive training and validation losses. Crucially, it recorded the highest testing accuracy of 92.7%, with precision and recall values of 92.61% and 92.58%, respectively, demonstrating its superior ability to generalize to unseen data. Furthermore, YOLOv8m-cls exhibited a faster training time of 1.036 hours comparing to YOLOv8l-cls and YOLOv8x-cls. This optimal balance of high testing accuracy, precision, recall and efficient training time positions YOLOv8m-cls as the most suitable choice for image classification tasks on the Kvasir dataset. Figure 3.3 illustrates training and validation losses per epoch, while figure 3.4 shows the top-1 accuracy per epoch for training the YOLOv8m-cls model on the Kvasir dataset.

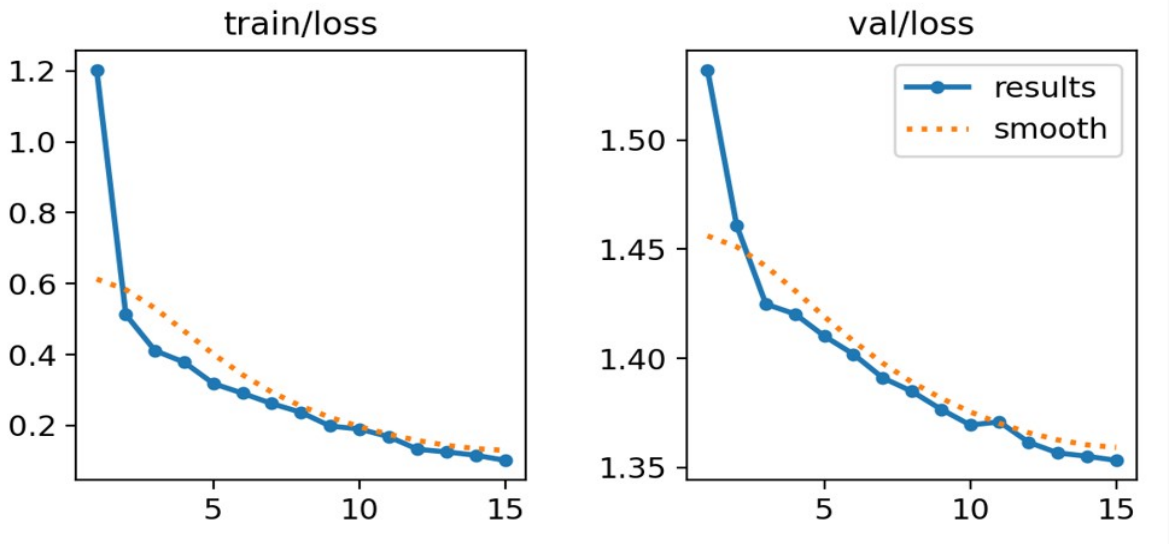

Figure 3.3: Training and validation losses per epochs for YOLOv8m-cls.

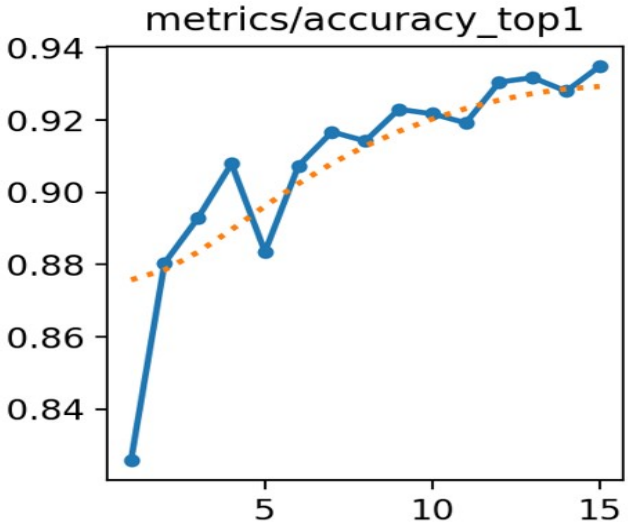

Figure 3.4: Accuracy top 1 per epochs for YOLOv8m-cls.

The Normalized Confusion Matrix demonstrated in figure 3.5 for YOLOv8m-cls in gastrointestinal diseases diagnosis shows remarkable performance on testing data for most classes, with high correct classification probability values ranging from 96% to 99% for dyed-lifted-polyps, normal-cecum, normal-pylorus, polyps and ulcerative-colitis. However, there is a noticeable confusion between esophagitis and normal z-line, with a non-negligible values of 0.29 and 0.08 in the off-diagonal cells, indicating that the model may struggle to distinguish between these classes due to the high visual similarity of the images within both classes; which makes the model's performance for esophagitis and normal z-line relatively lower with probabilities of 71% and 90% respectively. Overall, the model demonstrates promising results, but further improvements are needed to address the confusion between esophagitis and normal z-line.

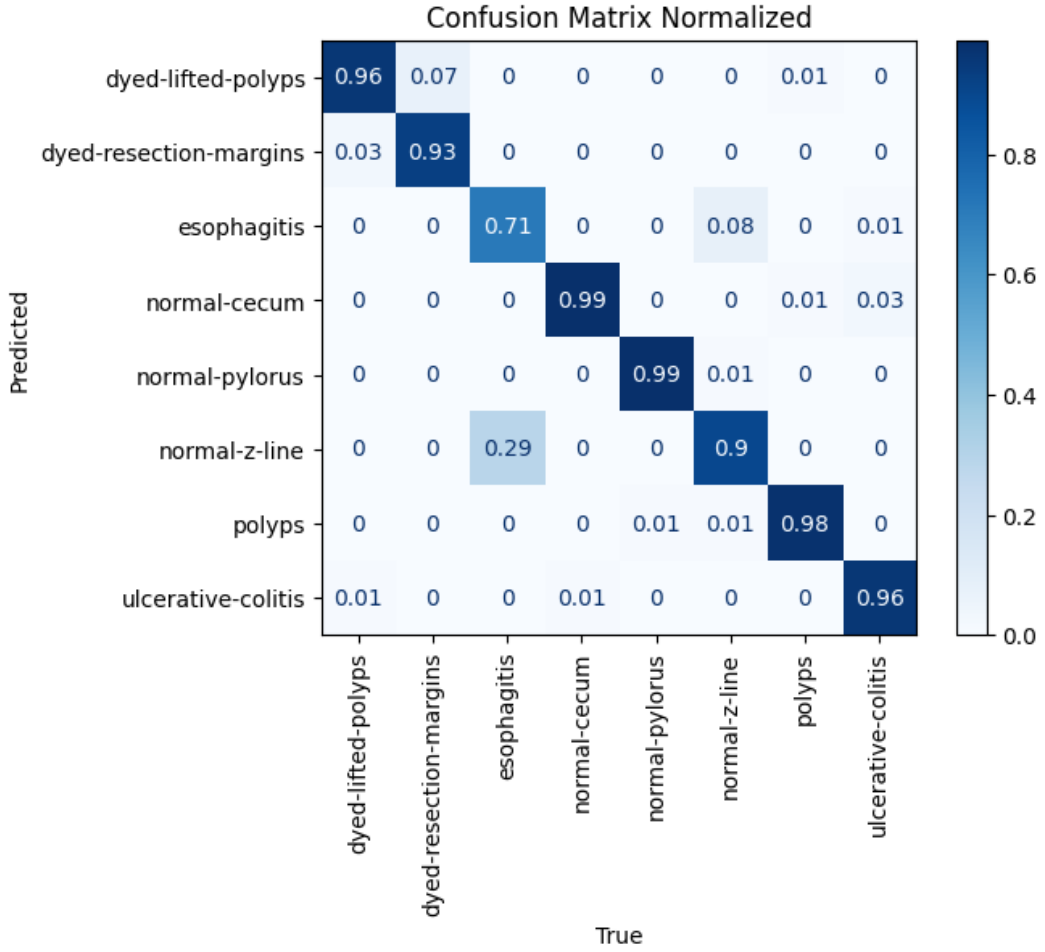

Figure 3.5: The Normalized Confusion Matrix for YOLOv8m-CLS in gastrointestinal diseases diagnosis.

3.6.2 Experiment 2: Performance Evaluation of YOLOv8m-cls on a 5-Class Kvasir Dataset

Following the evaluation of YOLOv8 variants on the full Kvasir dataset and concluding that YOLOv8m-cls is the best-performing variant, We now aim to evaluate its performance on a subset of the Kvasir dataset, consisting of only 5 classes, by eliminating the classes that cause confusion, to compare it with related works that used the same subset. As discussed in the related work section, previous studies have excluded 3 classes (esophagitis, normal Z-line, and dyed resection margins) from the Kvasir dataset due to their high visual similarity, which can lead to classification confusion. To align with these studies, we selected the following 5 classes: dyed lifted polyps, normal cecum, normal pylorus, polyps, and ulcerative colitis. We trained the YOLOv8m-cls model on this subset using the default hyperparameters provided by the YOLO framework. Specifically, the image size was set to 640, the batch size to 16, and the optimizer was set to auto, which selected AdamW with an initial learning rate of 0.000714 and momentum of 0.9. We also used the default augmentation and preprocessing techniques used by YOLOv8, as discussed in the data augmentation and preprocessing section. The model was trained for 15 epochs using a Tesla T4 GPU and then evaluated its performance on the testing data to assess its accuracy and effectiveness. The figure 3.6 illustrates a general structure

of this experiment.

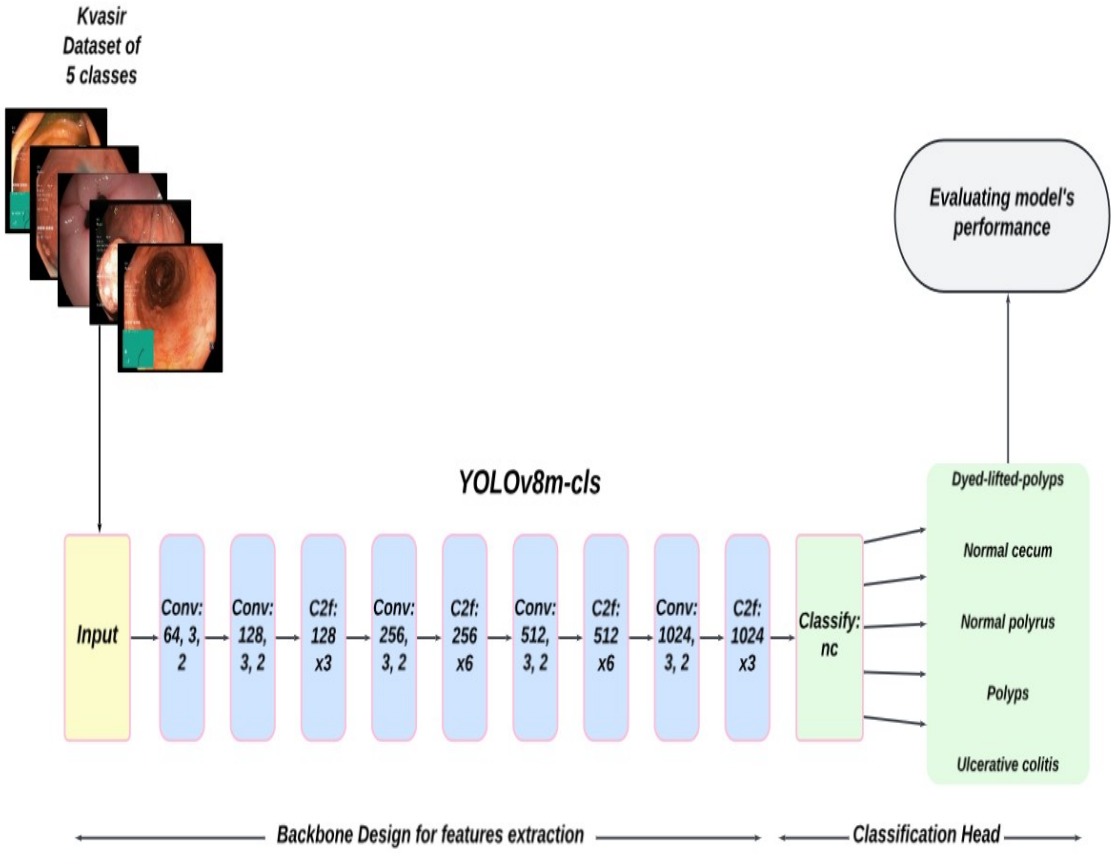

Figure 3.6: Structure of YOLOv8m-cls for gastrointestinal diseases diagnosis using 5-classes kvasir.

Results: After training our YOLOv8m-cls model on the 5-class subset of the Kvasir dataset, it achieved a top-1 accuracy of 97.78%. The model exhibited a training loss of 0.0396 and a validation loss of 0.9385 within an training time of 0.48 hours. The figure 3.7 presents the training and validation loss curves during the training process, while figure 3.8 represents the top-1 accuracy during the training process.

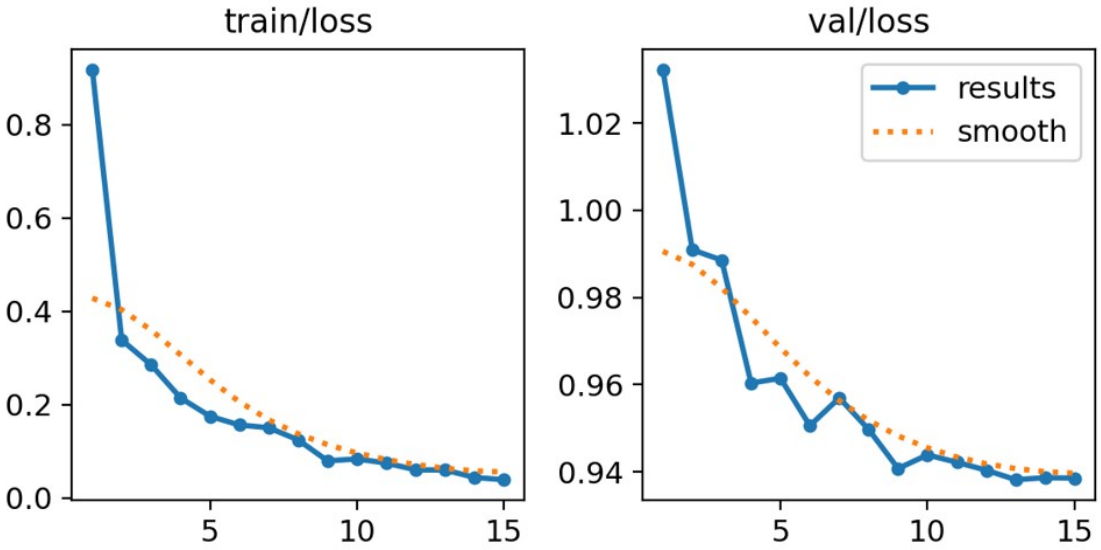

Figure 3.7: Training and validation loss per epochs.

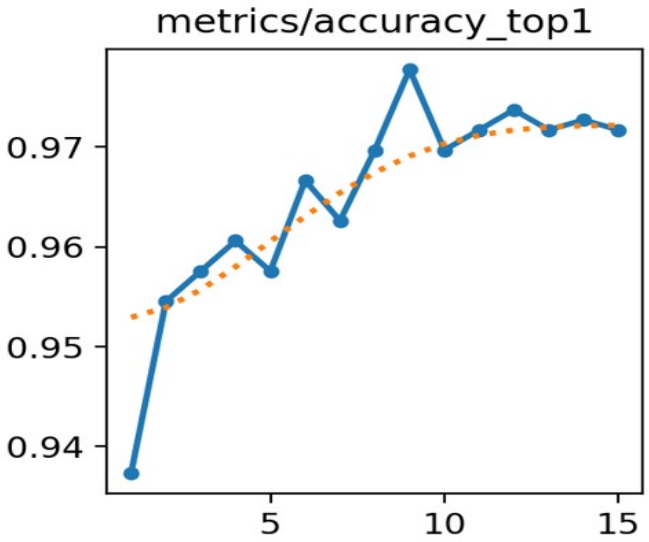

Figure 3.8: Top-1 accuracy per epochs.

Additionally, when evaluating the model's performance on the testing set, it achieved an accuracy of 97.39%, a precision of 97.34% and a recall of 97.24% on testing data. The figure 3.9 illustrates the normalized confusion matrix after evaluating our model on the testing set.

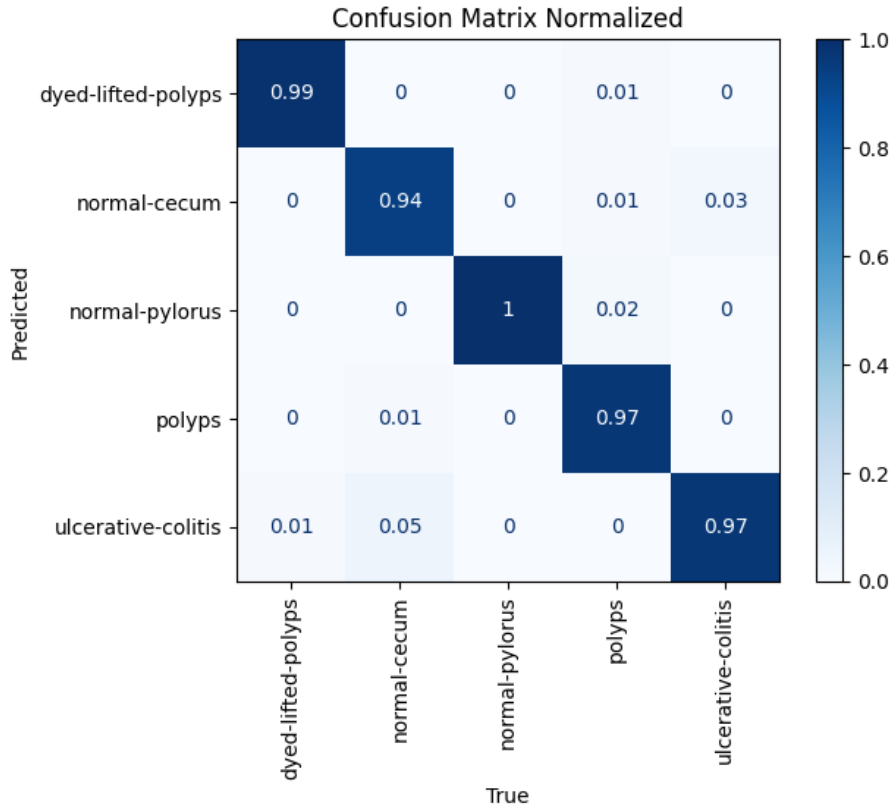

Figure 3.9: Normalized Confusion Matrix for YOLOv8m-cls on 5-Class Kvasir Dataset

Discussion: In the study conducted by Mousa Alhajlah et al. [29], GoogleNet, ResNet-50, and AlexNet demonstrated accuracies of 96.70%, 96%, and 97% respectively, with training times of 608, 693, and 100 minutes. Our YOLOv8m-cls model, trained on the identical 5-class subset of the Kvasir dataset, achieved a higher top-1 accuracy of 97.78% whithin shorter training time of 29 minutes and 16 seconds. Additionally, our model has a better performance on unseen data, achieving a higher testing accuracy of 97.39%, precision of 97.34%, and recall of 97.24% compared to the best performing AlexNet's validation accuracy of 94.88%. Furthermore, our model remained competitively performant with other models while also being more efficient on unseen data.

Following the evaluation of YOLOv8m-cls on the subset containing only 5 classes, we observed that it demonstrated good performance compared to previous studies. We now aim to improve its performance on the full Kvasir dataset. To further enhance its performance and eliminate confusion between specific classes, we aim to implement several enhancements in the following experiments.

3.6.3 Experiemt 3: Hyperparameter Tuning of YOLOv8m-cls for Enhanced GI Diseases Classification

After determining that the YOLOv8m-cls variant is the best-performing model for gastrointestinal disease classification from our previous experiments, we now seek to optimize its hyperparameters using Ray Tune by exploring various training hyperparameters to optimize model performance and enhance generalization on the full Kvasir dataset. This process will help us identify the optimal training configurations, thereby improving

the model's accuracy and robustness.

To conduct the hyperparameter tuning experiment, we utilized the Ray Tune library along with the HyperOpt search algorithm. The search space for hyperparameters included the learning rate (lr0), momentum, and choice of optimizer. The HyperOptSearch algorithm, initially configured with a set of parameters and the optimization metric (top1_accuracy), efficiently explored the hyperparameter space. After each training trial, the algorithm dynamically updated the search space based on past trial results, leveraging Bayesian optimization to iteratively improve performance. This involved adjusting the next set of hyperparameter values for lr0 (learning rate), momentum, and optimizer choice while respecting the tune value range .

To manage computational resources effectively, we employed the ConcurrencyLimiter to restrict the maximum number of concurrent trials to 4. The ASHAScheduler dynamically adjusted resource allocation based on trial performance, ensuring efficient exploration while maximizing resource utilization.

Finally, the objective function trained the YOLOv8 model with specified hyperparameters, utilizing the Ultralytics library for training on the Kvasir dataset for 15 epochs with a batch size set to 16. The following table (tab 3.3) represents the hyperparameter configurations of all training trials:

Model	Optimizer	lr0	Momentum	Batch size
YOLOv8m-cls-1	AdamW	0.00055351	0.859626	16
YOLOV8m-cls-2	AdamW	0.0.00561699	0.766142	16
YOLOV8m-cls-3	SGD	0.000766878	0.866162	16
YOLOV8m-cls-4	AdamW	0.000148792	0.668723	16
YOLOV8m-cls-5	SGD	0.00372	0.73114	16
YOLOV8m-cls-6	Adam	1.11266e-5	0.915282	16
YOLOV8m-cls-7	SGD	0.01	0.9	16
YOLOV8m-cls-8	SGD	0.00279	0.673887	16
YOLOV8m-cls-9	Adam	0.0323752	0.661203	16

Table 3.3: Hyperparameters tunning configurations for 9 training trials.

Results: We tracked the metrics of each training trial, including accuracy and loss while considering the training time, to identify the best-performing model. The table 3.4 represents the results obtained for each training trial.

Model	Training loss	Validation loss	Accuracy_top1	Training time (s)
YOLOv8m-cls-1	0.09478	1.3433	94.3644%	1670.66
YOLOV8m-cls-2	0.09575	1.34615	94.05%	1654.73
YOLOV8m-cls-3	0.3709	1.412	90.92%	1505.71
YOLOV8m-cls-4	0.1409	1.35383	93.3%	1661.73
YOLOV8m-cls-5	0.27	1.3768	92.7%	1533.72
YOLOV8m-cls-6	0.3353	1.3954	91.79%	1699.83
YOLOV8m-cls-7	0.1141	1.3565	93.2%	1578.62
YOLOV8m-cls-8	0.3181	1.3914	92.235%	1567.21
YOLOV8m-cls-9	0.2726	1.3879	91.86%	1560

Table 3.4: Evaluation metrics using Hyperparameters tunning for 9 training trials.

Discussion: After assessing multiple hyperparameter configurations, the top-performing model was identified as YOLOv8m-cls-1, achieving an accuracy of 94.3644%, a training loss of 0.09478 and a validation loss of 1.3433, completed in 1670.66 seconds. This configuration, utilizing the AdamW optimizer with a learning rate of 0.00055351 and a momentum of 0.859626, notably outperformed other configurations, including the default hyperparameters configuration from the previous experiment, which yielded an accuracy of 93.488%. Other notable configurations included YOLOv8m-cls-2 with 94.05% accuracy but slightly higher losses and a similar training time, and YOLOv8m-cls-7 achieving 93.2% accuracy with a lower training time of 1578.62 seconds but a higher losses. Models like YOLOv8m-cls-3 and YOLOv8m-cls-6 displayed accuracies of 90.92% and 91.79%, respectively, highlighting the impact of different optimizer choices on model performance and convergence. The combination of AdamW with a moderate learning rate of 0.00055351 and high momentum of 0.859626 provided a balance between learning speed and stability, leading to better convergence and generalization on the gastrointestinal disease classification task.

3.6.4 Experiment 4: Mixture of experts using YOLOv8m-cls for enhaced GI diseases diagnosis

Mixture of Experts (MoE) is an ensembling technique based on divide and conquer principle where the problem space is divided between experts[80]. In the field of deep learning, Mixture of Experts is a prominent technique that involves utilizing multiple expert networks to partition a problem space; by deviding the dataset into local subsets, with each model or "expert" trained on a specific subset, and a gating network that dynamically selects the most appropriate expert for each input. This approach is used to solve a complex problem by diving it into several simpler sub-problems which have simpler solutions. These individual solutions to simpler problems are then combined to produce a solution to the original complex problem[80].

In response to the confusion between esophagitis and normal Z-line identified in the

experiment 1, a Mixture of Experts approach was implemented to resolve this issue and reduce confusion. The methodology involves deviding the dataset into two subsets, each consisting of four classes, thereby ensuring that classes contributing to the confusion are placed in separate subsets. The first subset comprises dyed resection margins, esophagitis, polyps, and ulcerative colitis, intended for training the first expert. Conversely, the second subset comprises dyed lifted polyps, normal polyps, normal cecum, and normal Z-line classes, designated for training the second expert. Both experts were trained for 15 epochs using the best hyperparameter configuration obtained from Experiment 3, featuring the AdamW optimizer, an initial learning rate of 0.00055351, momentum of 0.859626 and a batch size of 16.

Subsequently, a third YOLOv8m-cls model, termed the gate, is trained using precisely the same hyperparameters configuration to select the most suitable expert for each input. This gate model is trained on the two subsets without class labeling, instead we mixed the images inside each subset without class labeling in order to create only 2 classes for this training, so our gate can decide which model to use for prediction corresponding to the subsets used for training the two experts.

Upon completion of experts and gate models training, we constructed our proposed model by combining the two experts and the gate. Input images are passed through the gate model to determine the appropriate expert for classification, with the selected expert making the final image class prediction.

Results: After training the expert model 1 on the first subset, it achieved a top 1 accuracy of 99.0%, a training loss of 0.02843 and a validation loss of 0.76166. Figure 3.10 below shows the training and validation losses, in addition to the top 1 accuracy per epochs for expert 1, where figure 3.11 illustrates the confusion matrix for expert 1 performance.

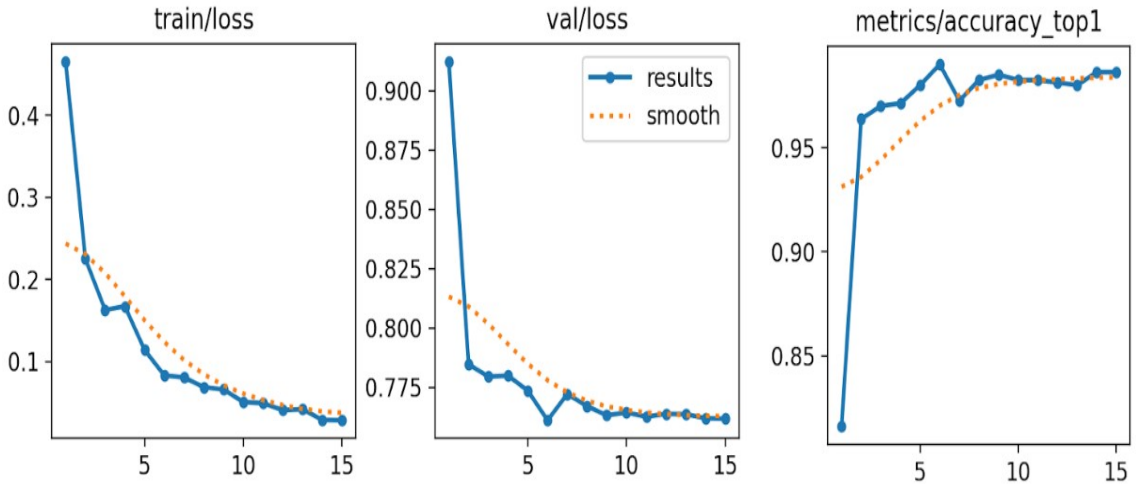

Figure 3.10: Losses and top 1 accuracy per epochs for expert 1 training.

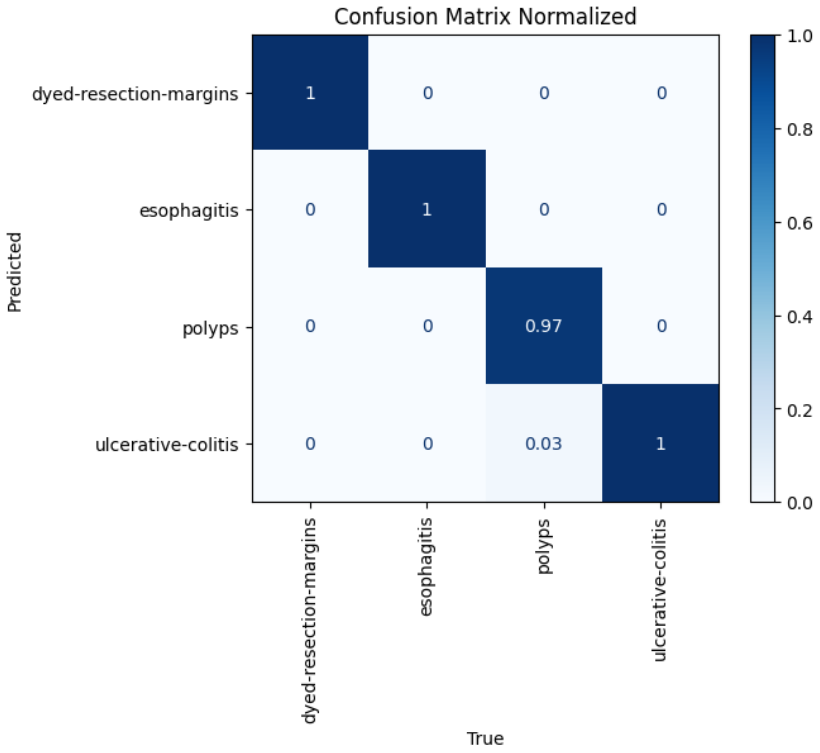

Figure 3.11: Confusion matrix for expert 1.

Expert 2 achieved a top1 accuracy of 100% with training and validation loss of 0.01226 and 0.74514 respectively. Figure 3.12 below represents the evaluation metrics per training epochs, where figure 3.13 shows the confusion matrix for expert 2.

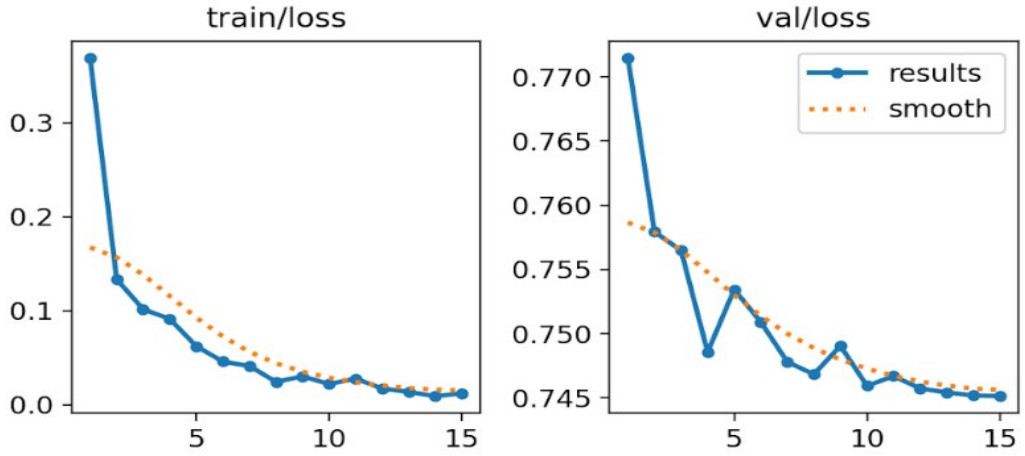

Figure 3.12: Training and evaluation losses per epochs for expert 2 training.

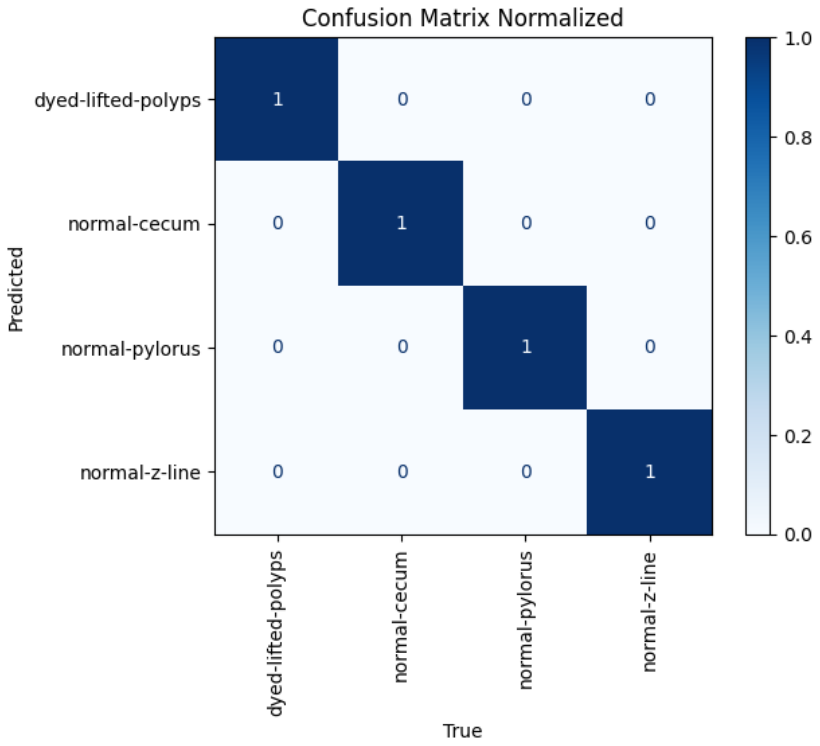

Figure 3.13: Confusion matrix for expert 2.

The gate model acheived a top 1 accuracy of 93.25%, a tarining loss of 0.09991 and a validation loss of 0.38289. Figures 3.14 and 3.15 show the evaluation metrics of the gate model per epochs during training, while figure 3.16 represents the confusion matrix of the gate model for choosing the appropiate model.

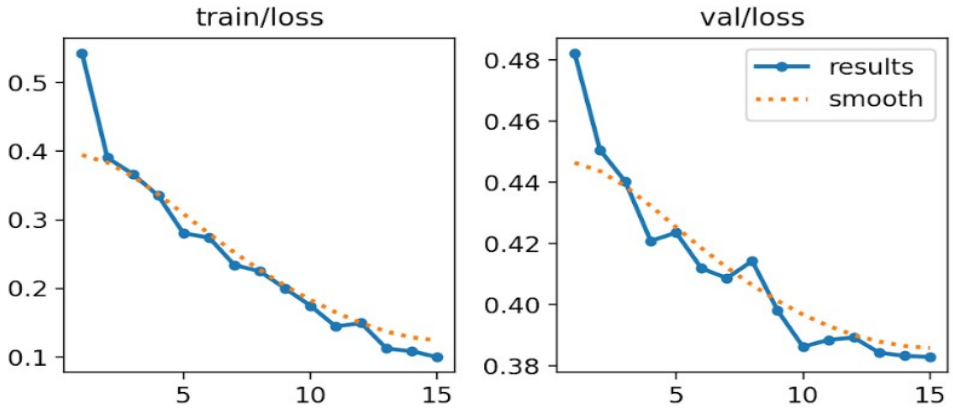

Figure 3.14: Training and validation losses per epochs for gate model training.

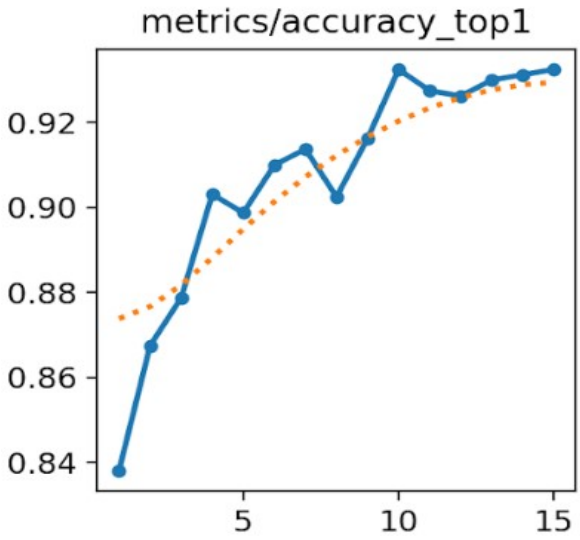

Figure 3.15: Top 1 accuracy per epochs for gate model training.

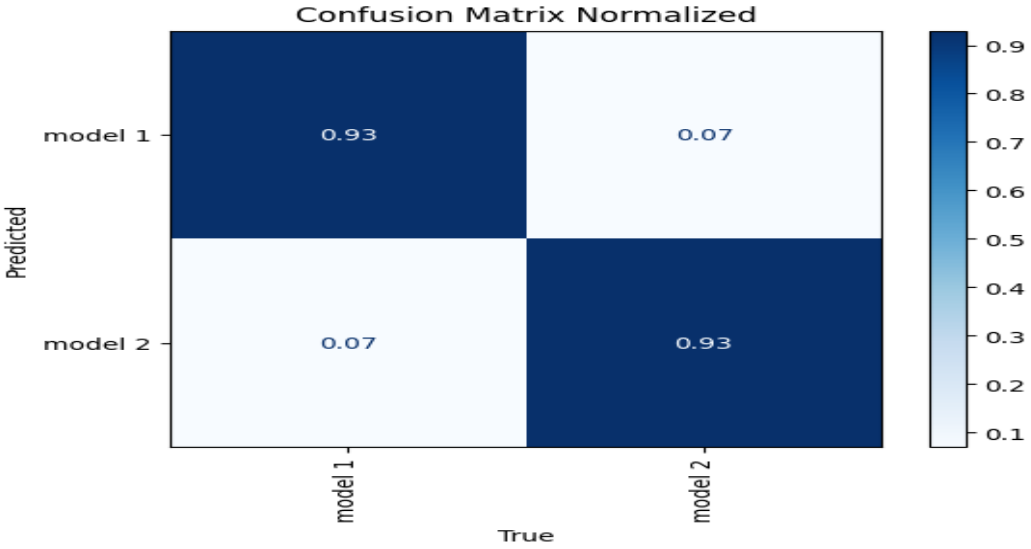

Figure 3.16: Confusion matrix for gate model's performance.

Finally, After constructing our model using the Mixture of Experts technique by combining the experts and the gate model, we evaluated its performance on our testing data. The overall model achieved an accuracy of 96.25%, a precision of 96.28% and a recall of 96.25% on the testing data, with an inference time of 20.6 ms for each image. The figure 3.17 illustrates both the confusion matrix and the normalized confusion matrix, showcasing the model's performance on the testing data.

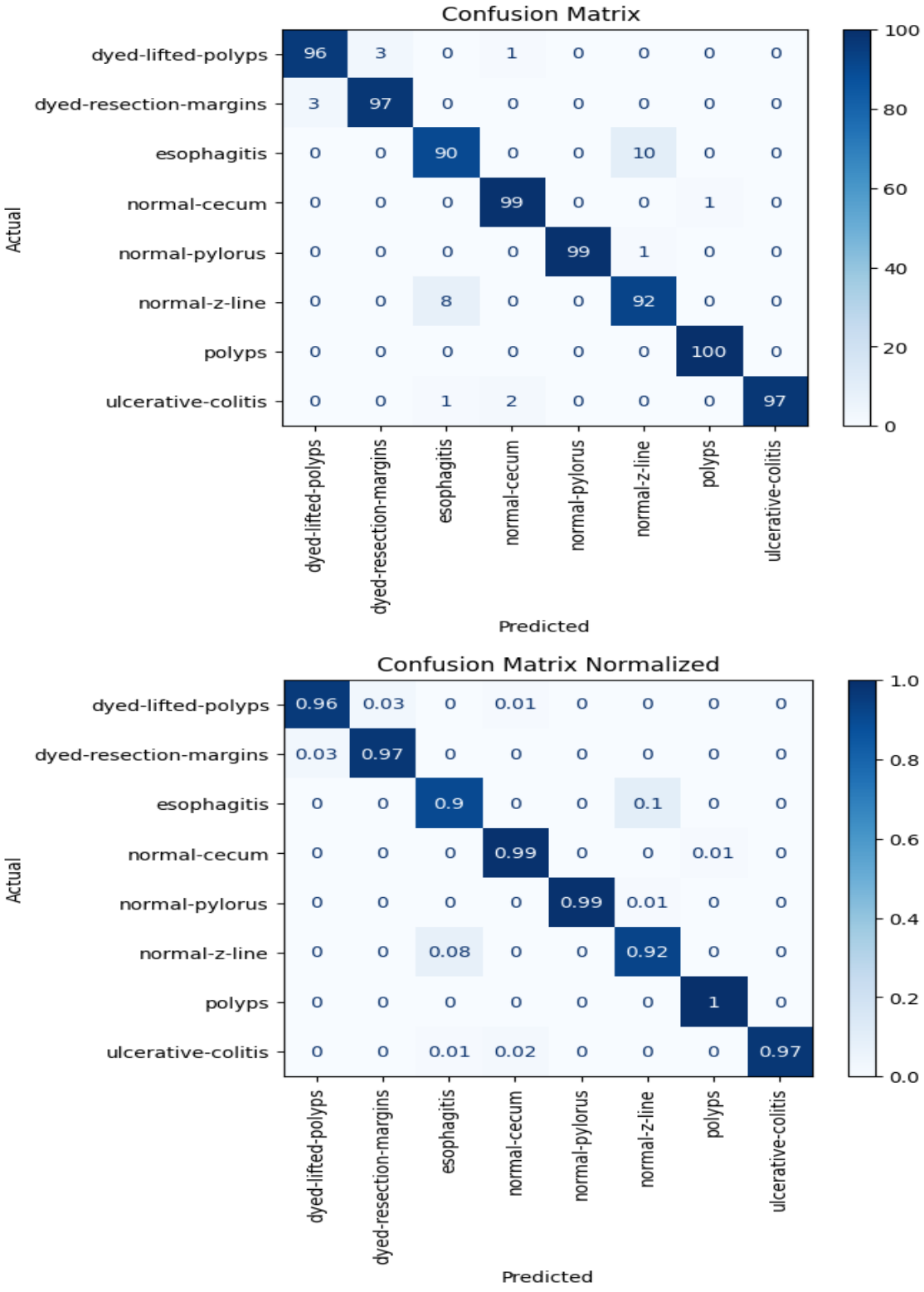

Figure 3.17: confusion matrices of our proposed model for gastrointestinal diseases diagnosis.

Discussion: Our experts models achieved very high performance after splitting our dataset into two subsets and training each expert on a subset, significantly minimizing confusion between classes. This performance make the expert appropriate to construct our proposed model using Mixture of Expert technique. The model we suggested showed a significant improvement attaining a testing accuracy of 96.25%, a precsion of 96.28% and a recall of 96.25%; which is a 4% increase in testing accuracy, precision, and recall compared to the best yolov8-cls model from experiment 2 that achieved a testing accuracy of 92.7% , precision and recall of 92.68% and 92.58% respectively.

The enhancement successfully reduced the confusion which was seen in experiment 1 between esophagitis and the normal Z-line from 0.29 to 0.1 misclassification probability, also between the other classes such as dyed-lifted-polyps and dyed-resection-margins from 0.07 to 0.03. This minimization in classification errors and improvement in performance metrics highlights the effectiveness of the Mixture of Experts approach in enhancing model performance.

3.7 General Discussion:

The performance of YOLOv8-cls models consistently demonstrated considerable advancements in gastrointestinal disease classification using the Kvasir dataset throughout the experiments.

The initial experiment expanded the scope to the full Kvasir dataset, where YOLOv8m-cls showed the best generalization to unseen data with a testing accuracy of 92.7% and a considerable training time, making it the best variant to use in the next experiments. In the second experiment, the YOLOv8m-cls model outperformed other models that are mentioned in related work like GoogleNet, ResNet-50, and AlexNet, achieving a testing accuracy of 97.39% while requiring less training time. This established a strong foundation for our model's capabilities.

In the third experiment, hyperparameter tuning further optimized the YOLOv8m-cls model, improving its accuracy to 94.3644%. This tuning involved adjusting learning rate and momentum, showcasing the model's potential for even better performance with the right configurations.

The final model; constructed using Mixture of Experts technique to address classification confusion and enhace our model's performance, achieved a testing accuracy of 96.25% a precision of and a recall of 96.25%. Attaining a 4% increase over the best result from the first experiment. This improvement demonstrated the strong performance of our model and significantly reduced classification errors, particularly between challenging classes such as esophagitis and normal Z-line.

Table 3.5 represents a comparison of our used approaches with the previous studies on both 8-classes and 5-classes kvasir datasets.

Related studies	Used Approach	Dataset	Training time	Accuracy (%)	Recall (%)	Precision (%)
Our final proposed model	**Yolov8m-cls + Hyperparameter tuning + MEO**	**8-classes KvasirV2**	**-**	**96.25**	**96.25**	**96.28**
Our initial model	**Yolov8m-cls**	**5-classes Kvasir subset**	**29 min**	**97.39**	**97.34**	**97.24**
Mosleh Hmoud Al-Adhaileh et al [30].	GoogleNet	5-classes Kvasir subset	608 min 21 sec	96.70	96.60	-
Mosleh Hmoud Al-Adhaileh et al [30].	AlexNet	5-classes Kvasir subset	100 min 37 sec	94.88	96.80	-
Mosleh Hmoud Al-Adhaileh et al [30].	ResNet	5-classes Kvasir subset	693 min 25 sec	97.00	94.80	-
Konstantin Pogorelov et al [31].	-	8-classes KvasirV1	-	95.7	82.6	82.9
Andrea Asperti et al [32].	Ensemble of Inception + finetuning + data augmentation	8-classes KvasirV1	-	91.5	91.5	91.5

Table 3.5: Comparison of the performance of our proposed approach with the previous studies

3.8 Summary:

This chapter outlines a sequence of experiments designed to develop a model using YOLOv8-cls and optimize its performance for the purpose of classifying gastrointestinal diseases, utilizing the Kvasir dataset as the basis for training and evaluation. Overall, the experiments highlighted the effectiveness of YOLOv8-cls for gastrointestinal disease classification in a timely manner, with each step improving the model's accuracy, precision, and recall, ultimately achieving a robust and efficient diagnostic tool.

General Conclusion

This project aimed to develop a robust and efficient framework for classifying gastrointestinal diseases using the Kvasir dataset.

To achieve this, we began with an overview of the digestive system, covering the anatomy of its various parts and identifying key anatomical landmarks. We discussed different gastrointestinal diseases and polyp removal cases. We then described diagnostic methods and procedures including upper endoscopy, colonoscopy, and capsule endoscopy. This overview was essential to understand the cases described in the Kvasir dataset, which contains 8,000 images divided equally into eight classes. These classes encompass three anatomical landmarks (normal z-line, normal pylorus, and normal cecum), three diseases (esophagitis, ulcerative colitis, and polyps), and two polyp removal classes (dyed lifted polyps and dyed resection margins).

With a solid grasp of the dataset, we proceeded to describe the model used in this project, YOLOv8-cls. We provided a comprehensive history of the YOLO (You Only Look Once) model, tracing its development from the initial version to the latest YOLOv8. We also described its architecture, its different variants and hyperparameters. We then developed a YOLOv8-cls model for gastrointestinal disease classification.

Our methodology involved training all YOLOv8 variants on the Kvasir dataset to identify the best-performing model for further enhancement. This initial model exhibited significant confusion between specific classes. To address this, we optimized the model through hyperparameter tuning and constructed a final model using the mixture of experts approach. Our final enhanced model demonstrated significant performance improvements by minimizing class confusion and achieving a testing accuracy of 96.25%, precision of 96.28% and recall of 96.25%.

Our contributions in this work include developing a model; using YOLOv8-cls and the mixture of experts technique, that surpassed existing approaches in terms of training metrics and training time with a notable testing performance with a significant inference time, while minimizing confusion between classes that have high visual similarity. This demonstrates the superior efficacy and robustness of our approach in gastrointestinal disease classification.

For future development, we intend to implement several enhancements to further improve our model. Initially, we plan to leverage a larger dataset, such as HyperKvasir, which includes more detailed cases to enhance the robustness of our model. Additionally, we aim to improve model performance by modifying the architecture of YOLOv8. Specifically, we plan to enhance the model's feature extraction capabilities by modifying the C2f and convolutional blocks.

Finally, this project successfully developed a state-of-the-art model for classifying gastrointestinal diseases using the Kvasir dataset. This work has significant implications for improving diagnostic accuracy and efficiency in medical imaging.

Bibliography

[1] National Institute of Diabetes and Digestive and Kidney Disease. *Your Digestive System & How it Works*. Accessed on February 11, 2024. URL: https://www.niddk.nih.gov/health-information/digestive-diseases/digestive-system-how-it-works.

[2] StockIllustration. *Vector Illustration of a 3d Labeled Diagram of the Human Digestive System, Digestive Tract, Alimentary Canal*. Online. Available: https://atstockillustration.com/design/vector-illustration-of-a-3d-labeled-diagram-of-the-human-digestive-system-digestive-tract-alimentary-canal-by-atstockillustration-7968.

[3] *Anatomical Landmarks*. Accessed on February 13, 2024. URL: https://datasets.simula.no/kvasir.

[4] Konstantin Pogorelov et al. "KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection". In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. MMSys'17. Taipei, Taiwan: ACM, 2017, pp. 164–169. ISBN: 978-1-4503-5002-0. DOI: 10.1145/3083187.3083212.

[5] The Editors of Encyclopædia Britannica. *pylorus*. Accessed on February 13, 2024. URL: https://www.britannica.com/science/pylorus.

[6] Melanie Haas. *What Are the Functions of the Cecum?* Accessed on February 15, 2024. 2018. URL: https://sciencing.com/functions-cecum-6809336.html.

[7] Shazia R. Chaudhry and Bruno Bordoni. "Anatomy, Thorax, Esophagus". In: *National Library of Medicine* (2023). Accessed on February 15, 2024. URL: https://www.ncbi.nlm.nih.gov/books/NBK482513/.

[8] Preet Kahai et al. "Anatomy, Abdomen and Pelvis: Large Intestine". In: *National Library of Medicine* (2023). Accessed on February 15, 2024. URL: https://www.ncbi.nlm.nih.gov/books/NBK470577/.

[9] *Gastrointestinal Diseases*. Accessed on February 17, 2024. URL: https://my.clevelandclinic.org/health/articles/7040-gastrointestinal-diseases.

[10] Rui Wang et al. "Global, Regional, and National Burden of 10 Digestive Diseases in 204 Countries and Territories from 1990 to 2019". In: *National Library of Medicine* (2023). Accessed on February 18, 2024. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10088561/.

[11] Catiele Antunes and Ashish Sharma. "Esophagitis". In: *National Library of Medicine* (2023). Accessed on February 21, 2024. URL: https://www.ncbi.nlm.nih.gov/books/NBK442012/.

[12] Mayo Clinic. *Esophagitis*. Accessed on February 21, 2024. 2022. URL: https://www.mayoclinic.org/diseases-conditions/esophagitis/symptoms-causes/syc-20361224.

[13] Cleveland Clinic. *Esophagitis,Symptoms and Causes*. Accessed on February 22, 2024. URL: https://my.clevelandclinic.org/health/diseases/10138-esophagitis.

[14] Cleveland Clinic. *Esophagitis,Management and Treatment*. Accessed on February 22, 2024. URL: https://my.clevelandclinic.org/health/diseases/10138-esophagitis.

[15] Whitney D. Lynch and Ronald Hsu. *Ulcerative Colitis*. Accessed on February 23, 2024. 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK459282/.

[16] Cleveland Clinic. *Symptoms and Causes*. Accessed on February 25, 2024. URL: https://my.clevelandclinic.org/health/diseases/10351-ulcerative-colitis.

[17] Whitney D. Lynch and Ronald Hsu. "Ulcerative Colitis: Treatment / Management". In: *National Library of Medicine* (2023). Accessed on February 23, 2024. URL: https://www.ncbi.nlm.nih.gov/books/NBK459282/.

[18] Noam Shussman and Steven D. Wexner. "Colorectal Polyps and Polyposis Syndromes". In: *Gastroenterology report* 2.1 (2014). Accessed on February 25, 2024, pp. 1–15. URL: https://pubmed.ncbi.nlm.nih.gov/24760231/.

[19] National Institute of Diabetes and Digestive and Kidney Disease. *Symptoms & Causes of Colon Polyps*. Accessed on February 27, 2024. URL: https://www.niddk.nih.gov/health-information/digestive-diseases/colon-polyps/symptoms-causes.

[20] National Institute of Diabetes and Digestive and Kidney Disease. *Diagnosis of Colon Polyps*. Accessed on February 27, 2024. URL: https://www.niddk.nih.gov/health-information/digestive-diseases/colon-polyps/diagnosis.

[21] National Institute of Diabetes and Digestive and Kidney Disease. *Treatment for Colon Polyps*. Accessed on February 27, 2024. URL: https://www.niddk.nih.gov/health-information/digestive-diseases/colon-polyps/treatment.

[22] *Polyp Removal*. Accessed on June 1, 2024. URL: https://datasets.simula.no/kvasir/.

[23] National Institute of Diabetes and Digestive and Kidney Disease. *Upper GI Endoscopy*. Accessed on March 1, 2024. URL: https://www.niddk.nih.gov/health-information/diagnostic-tests/upper-gi-endoscopy.

[24] National Institute of Diabetes and Digestive and Kidney Disease. *Colonoscopy*. Accessed on March 1, 2024. URL: https://www.niddk.nih.gov/health-information/diagnostic-tests/colonoscopy.

[25] PennMedicine. *Capsule Endoscopy*. Accessed on March 5, 2024. URL: https://www.pennmedicine.org/for-patients-and-visitors/find-a-program-or-service/gastroenterology/diagnostic-testing-and-procedures/endoscopy-procedures/capsule-endoscopy.

[26] *Capsule Endoscopy*. Online. Available: https://www.ypo.education/gastrointestinal/capsule-endoscopy-t659/video/.

[27] A. Esteva et al. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *Nature* 542.7639 (2017), pp. 115–118. DOI: 10.1038/nature21056.

[28] Laith Alzubaidi et al. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. In: *Journal of big Data* 8 (2021), pp. 1–74.

[29] Mousa Alhajlah et al. “Gastrointestinal Diseases Classification Using Deep Transfer Learning and Features Optimization”. In: *Comput. Mater. Contin* (2023). Received: 29 April 2022; Accepted: 04 August 2022. URL: https://cdn.techscience.cn/files/cmc/2023/TSP_CMC-75-1/TSP_CMC_31890/TSP_CMC_31890.pdf.

[30] Mosleh Hmoud Al-Adhaileh et al. “Deep learning algorithms for detection and classification of gastrointestinal diseases”. In: *Complexity* 2021 (2021), pp. 1–12.

[31] Konstantin Pogorelov et al. “A Comparison of Deep Learning with Global Features for Gastrointestinal Disease Detection”. In: (2017).

[32] Andrea Asperti and Claudio Mastronardo. “The Effectiveness of Data Augmentation for Detection of Gastrointestinal Diseases from Endoscopical Images”. In: *arXiv preprint arXiv:1712.03689* (2021).

[33] Xu Zhang et al. “Gastric precancerous diseases classification using CNN with a concise model”. In: *PloS one* 12.9 (2017), e0185508. DOI: 10.1371/journal.pone.0185508.

[34] Shanhui Fan et al. “Computer-aided detection of small intestinal ulcer and erosion in wireless capsule endoscopy images”. In: *Physics in Medicine & Biology* 63.16 (2018), p. 165001.

[35] Tsuyoshi Ozawa et al. “Automated endoscopic detection and classification of colorectal polyps using convolutional neural networks”. In: *Therapeutic Advances in Gastroenterology* 13 (2020). PMID: 32231710. DOI: 10.1177/1756284820910659. URL: https://doi.org/10.1177/1756284820910659.

[36] Juan R. Terven and Diana M. Cordova-Esparaza. “A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond”. In: *arXiv preprint arXiv:2304.00501* (2023).

[37] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: (2016), pp. 779–788.

[38] Joseph Redmon and Ali Farhadi. “YOLO9000: Better, Faster, Stronger”. In: (2016).

[39] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: (2017), pp. 7263–7271.

[40] *The History of YOLO Object Detection Models from YOLOv1 to YOLOv8*. Online. Available: https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/.

[41] Chien-Yao Wang Alexey Bochkovskiy and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In: *arXiv preprint arXiv:2004.10934* (2020).

[42] Ultralytics. *YOLOv5 Architecture Description*. Accessed: 2024-05-16. URL: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/.

[43] Chuyi Li et al. *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*. Meituan Inc, 2022.

[44] roboflow. *What's New in YOLOv8*. Accessed: 2024-05-16. URL: https://blog.roboflow.com/whats-new-in-yolov8/.

[45] GitHub user RangeKing. Online. Available: https://blog.roboflow.com/whats-new-in-yolov8/.

[46] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[47] Matthew Stewart. *Simple Introduction to Convolutional Neural Networks*. Online. Available: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac.

[48] Keiron O'Shea and Ryan Nash. "An Introduction to Convolutional Neural Networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[49] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. pmlr. 2015, pp. 448–456.

[50] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning". In: *Neural networks* 107 (2018), pp. 3–11.

[51] *Brief summary of YOLOv8 model structure*. 2023. URL: https://github.com/ultralytics/ultralytics/issues/189.

[52] Thotakura SaiRam Mupparaju Sohan and Venkata RamiReddy. "A Review on YOLOv8 and its Advancements". In: *International Conference on Data Intelligence and Cognitive Informatics*. Springer. 2024, pp. 529–545.

[53] Abhishek Kumar Pandey. *Adaptive Average Pooling Layer*. Online. Available: https://medium.com/@akp83540/adaptive-average-pooling-layer-cb438d029022.

[54] DebugAH. Online. Available: https://debugah.com/global-average-pooling-layers-for-object-localization-10972/.

[55] Harsh Yadav. *Dropout in Neural Networks*. Online. Available: https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9.

[56] Diego Unzueta. *Fully Connected Layer vs. Convolutional Layer: Explained*. Online. Available: https://builtin.com/machine-learning/fully-connected-layer.

[57] Bala Priya C. *Softmax Activation Function: Everything You Need to Know*. June 2023. URL: https://www.pinecone.io/learn/softmax-activation/.

[58] Ajitesh Kumar. *Mean Squared Error vs Cross Entropy Loss Function*. Online. Available: https://vitalflux.com/mean-squared-error-vs-cross-entropy-loss-function/.

[59] *YOLOv8 Architecture Detailed Explanation - A Complete Breakdown*. Available from: https://www.youtube.com/watch?v=HQXhDO7COj8.

[60] Ultralytics github. Accessed on June 2, 2024. URL: https://github.com/ultralytics/ultralytics.

[61] Ultralytics. *Hyperparameter Tuning Guide*. Accessed: 2024-05-23. URL: https://docs.ultralytics.com/guides/hyperparameter-tuning/.

[62] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[63] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[64] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[65] datarian. Online. Available: https://datarian.io/blog/grid-search-random-search.

[66] Bernd Bischl et al. "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 13.2 (2023), e1484.

[67] Joseph Rocca. *Ensemble methods: bagging, boosting and stacking*. Online. Available: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

[68] *A Gentle Introduction to Mixture of Experts Ensembles*. Apr. 2021. URL: https://www.coodingdessign.com/machine-learning/a-gentle-introduction-to-mixture-of-experts-ensembles/.

[69] Anna Van Deusen. *Python Popularity: The Rise of A Global Programming Language*. Jan. 2023. URL: https://flatironschool.com/blog/python-popularity-the-rise-of-a-global-programming-language/.

[70] *Ultralytics*. Accessed on May 10, 2024. URL: https://docs.wandb.ai/guides/integrations/ultralytics.

[71] *Ray tune*. Accessed on May 10, 2024. URL: https://docs.ray.io/en/latest/tune/index.html.

[72] Mosleh Hmoud Al-Adhaileh et al. "Deep learning algorithms for detection and classification of gastrointestinal diseases". In: *Complexity* 2021.1 (2021), p. 6170416. URL: https://onlinelibrary.wiley.com/doi/full/10.1155/2021/6170416.

[73] *Kvasir A Multi-Class Image-Dataset for Computer Aided Gastrointestinal Disease Detection*. Accessed on May 11, 2024. URL: https://datasets.simula.no/kvasir/.

[74] *Augmentation Settings and Hyperparameters*. Accessed on May 12, 2024. URL: https://docs.ultralytics.com/modes/train/#train-settings.

[75] Vaibhav Jayaswal. *Performance Metrics: Confusion matrix, Precision, Recall, and F1 Score*. Accessed on May 11, 2024. 2020. URL: https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262.

[76] Muhammad Nouman Noor et al. "Efficient Gastrointestinal Disease Classification Using Pretrained Deep Convolutional Neural Network". In: *Electronics* 12.7 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12071557. URL: https://www.mdpi.com/2079-9292/12/7/1557.

[77] *Accuracy vs. Precision vs. Recall in Machine Learning: What's the Difference?* URL: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall.

[78] Jason Brownlee. *5 Useful Loss Functions*. 2022. URL: https://machinelearningmastery.com/5-useful-loss-functions/.

[79] Ahmed Fawzy Gad. *Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall*. Accessed on May 11, 2024. 2020. URL: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/.

[80] Sai Krishna Kalyan. *Designing Mixture of Deep Experts*. Canada: Universite Laval, July 2017.