CHAPTER 1

INTRODUCTION

Gastrointestinal (GI) diseases encompass a variety of conditions affecting the digestive system, including the esophagus, stomach, intestines, and related organs. These diseases significantly impact global health, contributing to high morbidity, mortality, and healthcare costs. The causes of GI diseases are diverse, including infections, inflammation, genetic factors, and lifestyle choices. Common conditions include gastroesophageal reflux disease (GERD), peptic ulcers, inflammatory bowel disease (IBD), colorectal cancer, and liver diseases like hepatitis and cirrhosis.

Symptoms vary widely, such as abdominal pain, bloating, diarrhea, constipation, nausea, vomiting, and gastrointestinal bleeding. Diagnosing GI disorders involves clinical evaluations, lab tests, imaging, and often endoscopy. Endoscopy is crucial for diagnosing, sampling tissues, and guiding treatments. However, interpreting endoscopic images can be subjective, depending on the gastroenterologist's expertise, potentially leading to diagnostic inconsistencies and missed pathologies.

To improve diagnostic accuracy, artificial intelligence (AI) and machine learning (ML) are being explored for analyzing endoscopic images. These technologies could revolutionize gastroenterology by enhancing image interpretation, leading to better diagnoses and patient outcomes. This study aims to develop AI-driven models for early detection and classification of GI diseases using deep learning and advanced image processing, ultimately improving healthcare providers' diagnostic capabilities.

Early detection of diseases is vital for effective preventive healthcare, reducing morbidity and mortality, and alleviating healthcare burdens. Identifying diseases at an early stage allows timely intervention, optimizing treatment, and preventing complications .This not only strains healthcare resources but also impacts patient wait times and the timely initiation of treatment. The process may contribute to inefficiencies in healthcare systems, further affecting clinical outcomes. Early detection is particularly beneficial for diseases like cancer, cardiovascular disorders, and infections, significantly improving survival rates and reducing mortality.

Timely diagnosis also reduces healthcare costs by preventing the need for advanced disease management, lengthy hospital stays, and long-term care. Early detection facilitates efficient resource allocation and minimizes financial burdens on patients and healthcare systems. Moreover, early diagnosis allows for preventive measures to control disease progression, such as managing hypertension to prevent cardiovascular events or detecting diabetic retinopathy early to prevent vision loss.

Public health initiatives like screening programs and education campaigns play a crucial role in early detection, improving population-level health outcomes and reducing the societal disease burden.

1.1 OBJECTIVES

Deep learning architectures, such as ResNet50, MobileNetV2, and VGG16, are advanced tools in computer vision, particularly for image classification tasks. These convolutional neural networks (CNNs) automatically learn hierarchical representations of images, improving classification accuracy and efficiency.

ResNet50 is a powerful CNN known for handling complex image tasks through its deep network of 50 layers and residual connections, which help avoid the vanishing gradient problem. MobileNetV2 is optimized for mobile and embedded devices, using depthwise separable convolutions to maintain high performance with low computational demands. VGG16 is simpler but effective, consisting of 16 layers and known for its strong performance in tasks with limited variability.

The advantages of deep learning architectures include their automatic feature extraction, scalability, and capability to handle large datasets. These models can be pre-trained on vast datasets and fine-tuned for specific tasks through transfer learning, which accelerates training and enhances performance.

Preprocessing Endoscopy Images: Preprocessing is crucial for enhancing deep learning model performance in gastrointestinal (GI) disease prediction. Techniques like contrast adjustment, denoising, and image registration improve image clarity and feature extraction. Cropping and resizing standardize image dimensions, while data augmentation increases dataset diversity, helping the model generalize better. ROI (Region of Interest) extraction isolates relevant anatomical structures, improving focus and accuracy.

Model Performance Evaluation: Key metrics for evaluating models include accuracy, sensitivity, and specificity. Accuracy reflects overall prediction correctness, sensitivity measures true positive detection, and specificity assesses true negative identification. These metrics help gauge the model's effectiveness in classifying GI diseases from endoscopy images.

Dataset Partitioning: Datasets are divided into training, validation, and testing subsets. Training data helps the model learn patterns, validation data fine-tunes performance, and testing data assesses real-world applicability. This partitioning ensures robust model evaluation.

Training and Evaluation: Models like ResNet50, MobileNetV2, and VGG16 undergo training using optimization techniques like stochastic gradient descent (SGD) and loss functions such as cross-entropy. Post-training, models are evaluated using metrics and visual tools like confusion matrices and ROC curves. Statistical analyses may be conducted to compare performances, guiding model selection and further optimization.

Through iterative refinements, these models can be optimized for clinical use, enhancing their predictive capabilities and contribution to early disease detection and better patient outcomes.

1.2 PROBLEM STATEMENT

The project addresses the critical need for accurate and timely diagnosis of gastrointestinal (GI) diseases using endoscopy imaging. GI diseases, which include conditions such as gastroesophageal reflux disease (GERD), peptic ulcer disease, inflammatory bowel disease (IBD), and colorectal cancer, require early detection for effective treatment and management. However, the interpretation of endoscopy images is often challenging and subjective, leading to variability in diagnoses and the risk of missed detections.

Project Objectives

The primary objective of this project is to develop a predictive model capable of automatically analyzing endoscopy images to accurately classify them into various GI disease categories. By leveraging advancements in artificial intelligence and deep learning, the project aims to enhance the diagnostic capabilities of endoscopy imaging, ultimately improving clinical decision-making and patient outcomes.

Focus Areas

1. **Enhanced Diagnostic Accuracy**:

   Develop deep learning models tailored for analyzing endoscopy images. Train these models to discern subtle patterns indicative of different GI diseases. Create robust classification frameworks to categorize images into distinct disease classes.

2. **Addressing Subjectivity and Variability**:

   Mitigate the subjectivity and variability inherent in human interpretation of endoscopy images. Use deep learning algorithms to standardize and automate the diagnostic process. Train models on large annotated datasets to ensure objective and consistent diagnostic assessments.

3. **Early Detection of GI Diseases**:

   Emphasize early detection by developing predictive models capable of identifying abnormalities and lesions at incipient stages. Leverage deep learning to analyze subtle visual cues, enabling healthcare providers to initiate timely interventions. Align with proactive and preventive healthcare paradigms by focusing on early detection, leading to personalized treatment plans and improved patient outcomes.

Expected Outcomes

- **Improved Diagnostic Tools**:

  Creation of sophisticated models that provide gastroenterologists with valuable insights for precise diagnosis and treatment planning. Reduction in reliance on individual expertise, minimizing the risk of misdiagnosis.

- **Enhanced Patient Care**:
  Timely detection and intervention, improving the efficacy of therapeutic strategies.Reduction in disease progression and associated complications, enhancing overall patient health and well-being.

By integrating advanced deep learning techniques, this project aims to revolutionize GI disease diagnosis, paving the way for more accurate, consistent, and early detection, thereby significantly improving clinical outcomes in gastroenterology.

1.3 CHAPTER WISE SUMMARY

Chapter 2: System Analysis

In this chapter, we conduct a comprehensive evaluation of both the Existing System and our Proposed System. The Existing System relies heavily on manual interpretation of endoscopic images for diagnosing gastrointestinal disorders. This manual approach introduces subjectivity, leading to potential inaccuracies and inconsistencies in diagnosis. Recognizing these limitations, our Proposed System is designed to automate the diagnostic process using advanced deep learning and machine learning techniques. This automation aims to enhance diagnostic accuracy and consistency, addressing the shortcomings of manual interpretation. Through detailed use case analysis, we identify the essential functionalities and interactions required for the proposed system, laying a robust foundation for the subsequent requirements specification.

Chapter 3: System Design

Chapter 3 outlines the detailed design of our Proposed System, focusing on the development of algorithms and methodologies across various phases of the diagnostic process. These phases include image preprocessing, feature extraction, and disease classification.This not only strains healthcare resources but also impacts patient wait times and the timely initiation of treatment. The process may contribute to inefficiencies in healthcare systems, further affecting clinical outcomes. We adopt a modular design approach, dividing the system into distinct modules that handle specific aspects of the diagnostic pipeline. This modularity facilitates ease of implementation, maintenance, and scalability. The design incorporates state-of-the-art techniques in image analysis and machine learning, ensuring the robustness and efficiency of the system.

Each module is meticulously designed to perform its designated function, contributing to the overall effectiveness of the diagnostic process.

Chapter 4: System Implementation

This chapter delves into the System Implementation phase, where theoretical concepts are transformed into practical applications. The implementation involves the development of each module according to the specified design, with a focus on code quality, scalability, and performance. Each module undergoes rigorous testing to validate its functionality and reliability. Comprehensive testing procedures are employed to ensure that the system meets the requirements for real-world clinical application. This chapter discusses the challenges faced during implementation and the solutions applied to overcome them, ensuring the system's robustness and efficiency.

Chapter 5: Conclusion and Future Scope

In the final chapter, we summarize the key findings and contributions of our project. We highlight the advancements made in automating the diagnosis of gastrointestinal disorders through endoscopic image analysis. The project demonstrates significant improvements in diagnostic accuracy and efficiency. We also explore potential future developments, such as enhancing system capabilities, integrating real-time analysis during endoscopic procedures, and incorporating additional diagnostic modalities. Our project represents a substantial advancement in gastrointestinal medicine, with promising implications for future research and clinical practice.

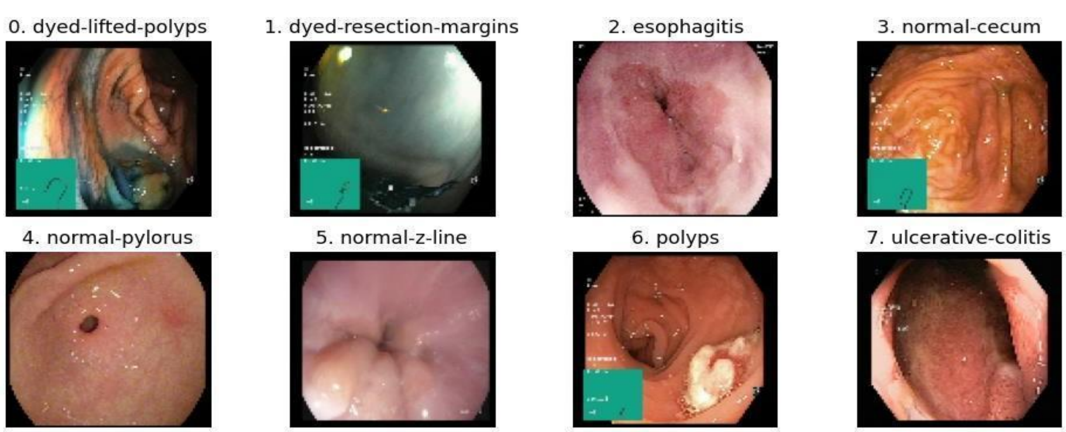

Fig 1.1 GI dataset

CHAPTER 2

SYSTEM ANALYSIS

System analysis is a crucial phase in any project, involving a detailed evaluation of its components, functionalities, and requirements. In the project "Prediction of Gastrointestinal Diseases from Endoscopy Images," system analysis plays a pivotal role. This not only strains healthcare resources but also impacts patient wait times and the timely initiation of treatment. The process may contribute to inefficiencies in healthcare systems, further affecting clinical outcomes. It encompasses key aspects necessary for developing and deploying predictive models for detecting gastrointestinal (GI) diseases through endoscopy images.

The initial focus of system analysis is on data collection and preprocessing. Comprehensive endoscopy image datasets from reliable sources are acquired, covering a wide range of GI diseases to ensure robustness and generalizability. Preprocessing techniques such as noise reduction, image enhancement, and data augmentation are applied to refine the datasets. These steps are vital for preparing high-quality input data, laying a solid foundation for model development and evaluation.

Following data preparation, model architecture selection is a critical step. Deep learning models such as ResNet50, MobileNetV2, and VGG16 are evaluated for their suitability in accurately classifying endoscopy images. The goal is to balance accuracy and computational efficiency, aligning with the project's objectives.

System analysis also involves the training and optimization of the selected models. Techniques like transfer learning and fine-tuning are used to adapt pre-trained models for GI disease prediction. Hyperparameter tuning, regularization, and optimization algorithms enhance model performance and generalization. Rigorous validation procedures, using metrics like accuracy, sensitivity, specificity, and AUC, ensure model robustness and reliability.

Deployment considerations are another essential aspect of system analysis. This includes evaluating computational infrastructure and integration with healthcare systems. Scalability, efficiency, and compatibility are prioritized to ensure seamless integration into clinical workflows. The ultimate aim is to enable real-time disease detection and decision support for healthcare professionals, improving diagnostic accuracy and patient outcomes in gastroenterology.

In summary, system analysis involves meticulous data collection and preprocessing to create a comprehensive and high-quality dataset. It focuses on selecting the most suitable deep learning architectures by balancing depth, computational efficiency, and performance. Training and optimization processes further refine the models using techniques like transfer learning, fine-tuning, and hyperparameter tuning. Finally, deployment considerations ensure the practical utility and integration of predictive models into clinical practice, contributing to enhanced patient care and clinical decision-making in gastroenterology.

2.1 EXISTING SYSTEM

Currently, gastrointestinal (GI) disease diagnosis from endoscopy images relies on manual interpretation by gastroenterologists. These professionals scrutinize images to identify abnormalities, lesions, and signs of disease, using their training and clinical judgment. This process involves analyzing the morphology, color, texture, and distribution of any abnormalities throughout the gastrointestinal tract. Despite its central role, manual interpretation has challenges, including variability in diagnostic accuracy due to individual differences in skill and experience, and cognitive biases.Grayscale conversion is used when color information is not essential, thereby reducing computational overhead. Data augmentation techniques such as rotation, flipping, translation, and zooming help increase the diversity of the training dataset. These transformations allow the models to generalize better by learning invariant features that are robust to variations in orientation, scale, and viewpoint.

The detection of subtle lesions, such as early-stage tumors or small mucosal changes, is particularly challenging. These lesions can be difficult to spot due to their small size, inconspicuous appearance, or lack of distinct visual features. Variability in interpretation and potential distractions can lead to missed or misinterpreted lesions, which may result in delayed diagnoses and treatment, affecting patient outcomes.

Additionally, manual interpretation is time-consuming, requiring thorough review of large volumes of images, which can delay diagnoses and treatments. This not only strains healthcare resources but also impacts patient wait times and the timely initiation of treatment. The process may contribute to inefficiencies in healthcare systems, further affecting clinical outcomes.

To address these challenges, advancements in artificial intelligence and computer-aided diagnosis are emerging to enhance the accuracy, efficiency, and speed of the diagnostic process, helping gastroenterologists in lesion detection and diagnosis.

2.2 PROPOSED SYSTEM

The proposed system integrates advanced deep learning models to revolutionize the prediction of gastrointestinal (GI) diseases from endoscopy images. Deep learning algorithms can automate and enhance disease diagnosis by identifying subtle visual cues associated with various GI pathologies, such as GERD, peptic ulcers, IBD, and colorectal cancer. These models analyze raw image data without manual annotation, improving diagnostic accuracy and reducing human error.

By uncovering complex patterns in the data, deep learning algorithms provide valuable decision support to healthcare professionals, helping prioritize cases and formulate personalized treatment plans. These models adapt over time through continuous learning, further optimizing diagnostic performance.

Pretrained models like ResNet50, MobileNetV2, and VGG16, typically trained on general image recognition tasks, serve as a foundation for GI disease prediction. Using transfer learning, these models are fine-tuned for endoscopy images, allowing efficient adaptation to the specific task of GI disease detection without needing extensive labeled datasets. Fine-tuning improves the model's performance by adjusting the parameters based on the target dataset.

Data preparation and preprocessing are vital for model effectiveness. Image enhancement techniques, such as adjusting brightness, contrast, and sharpness, improve visual clarity, helping models detect subtle features. Noise reduction techniques, including median filtering and Gaussian smoothing, help eliminate artifacts and improve the interpretability of images. Data augmentation, by generating new samples through transformations, increases dataset diversity and improves the model's ability to generalize, enhancing its robustness to real-world variations.

Together, these methods ensure high-quality data, improving the deep learning model's diagnostic accuracy and supporting efficient and accurate GI disease diagnosis.

2.3 USE CASE ANALYSIS

The use case analysis for our proposed system focuses on elucidating the key functionalities and interactions needed to ease the automated diagnosis of gastrointestinal (GI) disorders using endoscopic images. This analysis is instrumental in defining the system's requirements and guiding its design and implementation.

One of the primary use cases of our system involves the uploading and preprocessing of endoscopic images. Healthcare practitioners, upon performing endoscopic procedures, will upload the captured images to the system through a user-friendly interfaceGrayscale conversion is used when color information is not essential, thereby reducing computational overhead. Data augmentation techniques such as rotation, flipping, translation, and zooming help increase the diversity of the training dataset. These transformations allow the models to generalize better by learning invariant features that are robust to variations in orientation, scale, and viewpoint. Upon upload, the system will preprocess the images using techniques such as noise reduction and contrast enhancement to improve their quality and consistency. This step is crucial for ensuring that the images are suitable for analysis by the diagnostic models.

Another key use case is the feature extraction and disease classification process. Once the images are preprocessed, the system will employ deep learning models, such as convolutional neural networks (CNNs), to extract relevant features indicative of GI disorders. Furthermore, the system will incorporate a feedback mechanism to improve its diagnostic accuracy over time. As healthcare practitioners review and confirm the diagnostic results provided by the system, they will have the option to provide feedback on the accuracy of the diagnoses. This feedback will be used to continuously refine and update the deep learning models, thereby enhancing their performance and adaptability to new cases.

Additionally, the use case analysis encompasses functionalities related to result visualization and interpretation. Once the diagnostic process is completed, the system will present the results to healthcare practitioners in a clear and comprehensible manner. This may include displaying the predicted disease categories, confidence scores, and supporting evidence extracted from the endoscopic images. Healthcare practitioners can then review and interpret the results to make informed clinical decisions about patient management and treatment planning

2.4. REQUIREMENT SPECIFICATION

2.4.1 Functional Requirements:

Image Upload:

The system must enable healthcare practitioners to upload endoscopic images captured during diagnostic procedures such as gastroscopy and colonoscopy. This feature should offer an intuitive, user-friendly interface that minimizes complexity and ensures ease of use for medical personnel with varying levels of technical expertise. It should support a wide range of commonly used medical image formats, including JPEG, PNG, TIFF, and DICOM, to accommodate diverse equipment and imaging standards. Additionally, the upload process should include real-time validation to check image integrity, format compatibility, and resolution adequacy before the images proceed to the next stage of analysis.

Preprocessing:

Once the images are uploaded, the system should automatically initiate a comprehensive preprocessing pipeline to prepare them for diagnostic analysis. This preprocessing step is essential for enhancing image clarity, standardizing input data, and eliminating artifacts that may interfere with accurate interpretation. Specific tasks in this module should include:

- **Noise reduction** using filters such as median filtering or Gaussian smoothing to suppress random pixel-level fluctuations.
- **Contrast enhancement** and **sharpness adjustment** to improve visibility of fine structural details within the gastrointestinal tract.
- **Normalization** of pixel intensity values and image dimensions to maintain uniformity across the dataset, ensuring compatibility with the feature extraction and classification stages.

Feature Extraction:

Following preprocessing, the system should employ deep learning techniques—primarily convolutional neural networks (CNNs)—to automatically extract high-level features from the processed endoscopic images. These features should encapsulate complex visual patterns, textures, and anomalies that are indicative of gastrointestinal disorders such as gastritis, colitis, or malignancies. The extraction process must be designed to preserve spatial hierarchies and highlight pathological markers that are critical for accurate disease identification and differentiation.

Disease Classification:

Based on the extracted features, the system should classify each image into predefined categories of gastrointestinal diseases. This classification should be powered by trained deep learning models capable of recognizing nuanced differences between disease states. Categories may include but are not limited to gastritis, ulcerative colitis, Crohn's disease, gastrointestinal malignancies, and normal/healthy tissues. In addition to visual patterns, the model should optionally consider contextual metadata (e.g., patient history or procedural details) to refine classification outcomes and mimic real-world diagnostic reasoning.

Result Presentation:

The final diagnostic results should be displayed in a format that is both clear and clinically informative for healthcare professionals. This output should include:

- **Predicted disease categories** clearly labeled for each uploaded image.
- **Confidence scores**, representing the model's probability estimates for each potential diagnosis, allowing practitioners to assess the certainty of the prediction.
- **Supporting evidence**, such as heatmaps, saliency maps, or highlighted regions on the image, to visually indicate the areas that contributed most to the classification decision. These visual aids help build trust in the system and assist clinicians in correlating AI-generated insights with their own observations.

2.4.2. Non-Functional Requirements:

Performance:

The system must be capable of processing and analyzing endoscopic images with minimal latency to support real-time or near-real-time diagnostic workflows. Speed is crucial in clinical environments where timely decision-making can directly impact patient outcomes. This requires efficient image processing pipelines, optimized deep learning models, and fast response mechanisms that allow the system to deliver diagnostic results promptly without bottlenecks or delays.

Accuracy:

A core requirement of the system is high classification accuracy to ensure dependable diagnostic outcomes. The system should be rigorously trained and validated using diverse and representative datasets to reduce false positives and false negatives. Achieving high accuracy is essential to minimize misdiagnoses, enhance clinician confidence, and ensure that the AI model supports rather than hinders clinical decision-making. Continuous model evaluation and updates should be in place to maintain and improve accuracy over time.

Scalability:

The system architecture must be designed for scalability, ensuring that it can accommodate a growing number of users and an expanding volume of endoscopic image data. This includes support for horizontal scaling across cloud infrastructure, load balancing for high-demand periods, and the ability to integrate new datasets or model improvements without service disruption. A scalable system guarantees long-term sustainability and readiness for deployment across multiple healthcare facilities.

Reliability:

High system reliability is crucial in clinical settings, where any downtime could hinder diagnosis and treatment planning. The system should have robust failover mechanisms, automatic backups, and error recovery protocols to ensure continuous operation. Regular system monitoring, diagnostics, and maintenance routines should be in place to prevent unexpected failures and maintain a high standard of service availability.

Security:

To protect sensitive patient data, the system must implement stringent security measures in compliance with healthcare regulations such as HIPAA or GDPR. These should include:

- **End-to-end encryption** of data both in transit and at rest, safeguarding image files and patient information from unauthorized access.
- **User access control** through role-based authentication mechanisms to ensure that only authorized personnel can upload, view, or manage patient data.
- **Audit logging** to track all system activities related to data access, uploads, modifications, or deletions. These logs are critical for ensuring transparency, supporting security audits, and identifying potential breaches or misuse.

2.4.3. Hardware Requirements:

Computing Infrastructure:

To effectively support the demands of deep learning-based image analysis and preprocessing, the system requires robust and high-performance computing infrastructure. This includes powerful servers or cloud computing resources capable of running complex algorithms with speed and precision.

- **Multi-core processors** such as Intel Xeon or AMD Ryzen Threadripper are essential for parallelizing data preprocessing tasks and ensuring smooth system operations, especially when handling multiple images concurrently.
- The infrastructure must also include adequate RAM, typically 32GB or higher, to accommodate the high memory requirements associated with loading, processing, and feeding large image datasets into deep learning models without slowdowns or memory bottlenecks.

Storage:

Given the substantial volume of high-resolution endoscopic images generated in clinical practice, the system must incorporate efficient and scalable storage solutions.

- **Solid-State Drives (SSDs)** or **Network-Attached Storage (NAS)** systems should be used to store and retrieve image data rapidly, minimizing I/O latency during model training and inference.

- To support long-term usage and data growth, scalable storage architectures—capable of expanding dynamically as image volumes increase—are critical. These systems must also ensure high availability and redundancy to prevent data loss and ensure continuous access to medical image archives.

Graphics Processing Units (GPUs):

For the training and real-time inference of deep learning models, high-performance GPUs are indispensable.

- GPUs from the NVIDIA Tesla series (such as V100 or A100) or the GeForce RTX series are well-suited for handling the computational loads of convolutional neural networks used in image classification.

  The chosen GPUs must include CUDA support, enabling compatibility with popular deep learning frameworks like TensorFlow, PyTorch, and Keras, and ensuring maximum efficiency during both training and prediction phases. These units significantly reduce training time and accelerate inference, making them essential components in AI-powered diagnostic systems.

2.4.4. Software Requirements:

1. **Operating System:**

   Linux-based operating systems (e.g., Ubuntu, CentOS) for server environments, ensuring stability and compatibility with deep learning frameworks.

   Windows or macOS for user workstations, providing a more intuitive interface for clinicians.

2. **Deep Learning Frameworks:**

   TensorFlow: An open-source framework developed by Google, suitable for building and training deep learning models.

   Porch: A deep learning library developed by Facebook's AI research, known for flexibility and dynamic computation.

3. **Image Processing Libraries:**

   OpenCV: An open-source computer vision library providing various tools for image processing, feature extraction, and object detection.

   scikit-image: A Python library for image processing, offering algorithms for filtering, segmentation, and morphological analysis.

CHAPTER 3

SYSTEM DESIGN

3.1 DETAIL DESIGN

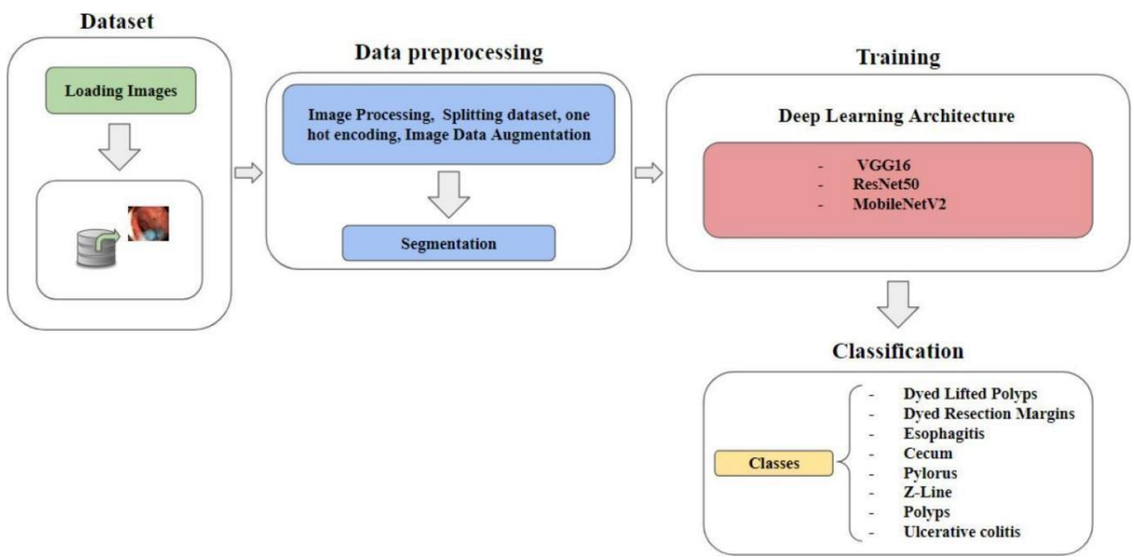

fig 3.1 System Architecture

In the Detailed Design phase as shown in fig 3.1, the architecture of deep learning models such as ResNet50, MobileNetV2, and VGG16 is carefully crafted to optimize performance and computational efficiency for the task of classifying gastrointestinal diseases from endoscopic images. Each model's architecture is designed to capture relevant features from medical images while balancing complexity and resource constraints. For instance, ResNet50 uses residual connections to alleviate the vanishing gradient problem, enabling better performance for deeper networks. MobileNetV2 is designed for efficient mobile and embedded systems, utilizing depthwise separable convolutions and inverted residuals to minimize computational demands while maintaining high accuracy. VGG16, known for its simple yet effective deep network design, uses stacked convolutional layers with small 3x3 filters to extract detailed features from images.Grayscale conversion is used when color information is not essential, thereby reducing computational overhead. Data augmentation techniques such as rotation, flipping, translation, and zooming help increase the diversity of the training dataset. These transformations allow the models to generalize better by learning invariant features that are robust to variations in orientation, scale, and viewpoint.

To ensure the models perform optimally, the design process also involves fine-tuning parameters such as kernel size, stride, padding, and activation functions. These architectural adjustments help the models process the endoscopic images efficiently while maintaining a high level of accuracy in predicting gastrointestinal disorders. Activation functions like ReLU are chosen for their ability to introduce non-linearity into the models, allowing them to learn complex relationships inherent in medical image data. These carefully selected parameters, such as kernel size and stride, are tuned to enhance the model's ability to capture both local and global spatial features in the images.

Data preprocessing and augmentation are critical aspects of the detailed design phase. Given the importance of image quality in accurate diagnosis, preprocessing steps such as resizing, normalization, and grayscale conversion are standardized. Resizing ensures that all images are of a uniform size, making them compatible with the input dimensions of the models. Normalization standardizes the pixel values, reducing variability caused by differences in lighting and contrast across images. Grayscale conversion is used when color information is not essential, thereby reducing computational overhead. Data augmentation techniques such as rotation, flipping, translation, and zooming help increase the diversity of the training dataset. These transformations allow the models to generalize better by learning invariant features that are robust to variations in orientation, scale, and viewpoint.

Effective data handling is essential to ensure that the endoscopic images are consistently processed throughout the training and evaluation phases. The dataset is divided into training, validation, and test sets, allowing for proper model evaluation and hyperparameter tuning. Emphasis is placed on datasets that capture a wide range of disease manifestations, anatomical regions, and imaging modalities, reflecting the diversity encountered in real-world clinical settings. This process helps avoid overfitting and ensures that the models generalize well to unseen data. Additionally, class imbalances are managed to prevent skewed predictions, and steps are taken to ensure data integrity and avoid leakage between the training and test sets.

The software infrastructure supporting the model's development and deployment is another crucial element of the design phase. Custom scripts are developed to automate various tasks such as data loading, preprocessing, and augmentation, ensuring consistency and reducing manual errors. Libraries like TensorFlow and PyTorch are used for model training and evaluation, providing the necessary tools for building, optimizing, and deploying deep learning models. Hyperparameter tuning is automated through modules that implement optimization techniques like grid search, random search, or Bayesian optimization, ensuring that the models achieve the best performance.

Once the models are trained and evaluated, deployment strategies are considered to integrate the system into clinical workflows. A web-based application allows healthcare practitioners to upload endoscopic images and receive diagnostic results in real-time. To support a growing number of users, cloud-based solutions are utilized for scalability. This ensures that the system can handle a large volume of image data without compromising performance or reliability. The models are also optimized for deployment through techniques such as model compression, which reduces their size without sacrificing accuracy, making them more efficient for use on resource-constrained devices.

In conclusion, the Detailed Design phase is fundamental in building a robust and efficient system for diagnosing gastrointestinal diseases from endoscopic images. By carefully designing model architectures, implementing rigorous data preprocessing and augmentation procedures, and developing a comprehensive software infrastructure, the system ensures reliable and accurate predictions. The end result is a system that is scalable, efficient, and capable of supporting clinical decision-making in healthcare environments.

3.2 DESIGN OF METHODOLOGY

In the Data Collection phase, the process begins by systematically acquiring endoscopy image datasets from reputable and clinically validated repositories. Strict criteria are used to select datasets, ensuring they cover a wide range of gastrointestinal diseases, including but not limited to gastroesophageal reflux disease (GERD), peptic ulcer disease, inflammatory bowel disease (IBD), and colorectal cancer. The selection criteria emphasize the reliability and credibility of the data source, dataset size, diversity, and quality of annotations.

Datasets are chosen based on their ability to provide a representative sample of endoscopic images that span a variety of gastrointestinal conditions and clinical scenarios. Emphasis is placed on datasets that capture a wide range of disease manifestations, anatomical regions, and imaging modalities, reflecting the diversity encountered in real-world clinical settings. The goal is to ensure the data set represents a comprehensive sample of conditions that will allow the trained models to learn from a broad spectrum of clinical scenarios.

Furthermore, attention is paid to the quality and completeness of annotations associated with the datasets. Proper labeling of pathological findings and anatomical structures within the endoscopy images is critical. Annotation quality is assessed based on criteria such as consistency, granularity, and domain accuracy, with the aim of minimizing discrepancies and ambiguities in the data. Emphasis is placed on datasets that capture a wide range of disease manifestations, anatomical regions, and imaging modalities, reflecting the diversity encountered in real-world clinical settings. Datasets that include accompanying metadata, such as patient demographics, clinical histories, and procedural details, are prioritized. This metadata enriches the contextual information and enhances the depth of the dataset, supporting better model training and evaluation.

By adhering to these stringent selection criteria, the method ensures that the endoscopy image datasets are representative and suitable for use in model training and evaluation. This rigorous approach establishes a strong foundation for building predictive models capable of accurately diagnosing gastrointestinal diseases from endoscopy images. These models, once developed, can significantly contribute to clinical decision-making, improving patient care and outcomes.

Preprocessing is an essential next step in preparing the dataset for model training. Image resizing is performed to standardize the dimensions of the endoscopic images, ensuring consistency and compatibility across the dataset. This step eliminates computational inefficiencies that might arise due to variations in image size and ensures uniformity. Additionally, normalization techniques are applied to scale pixel intensities to a common range, mitigating variations in lighting and contrast between images. By standardizing the pixel values, the preprocessing pipeline helps improve the consistency and comparability of image features, making the dataset more uniform for model training.

Data Augmentation techniques are used to further diversify the dataset and enhance the model's generalization ability. Methods such as rotation, flipping, zooming, and translation introduce variations to the dataset, allowing the model to learn invariant features and patterns across different orientations, scales, and viewpoints. Augmentation helps create a richer dataset that makes the model more robust and reduces the risk of overfitting by exposing it to a wide variety of data transformations.

To ensure the quality of the dataset, Quality Control measures are implemented to identify and resolve any outliers, artifacts, or data inconsistencies. These quality control steps may involve manual inspection, automated anomaly detection algorithms, or expert review to flag problematic samples. These efforts help maintain the integrity of the dataset, ensuring that only high-quality, reliable data is used in model training. This thorough process minimizes the risk of introducing errors or biases during training and evaluation, contributing to the development of a more accurate and reliable model.

In summary, the data collection and preprocessing phases involve a thorough and systematic approach to ensure that the datasets used for model training are of high quality, diverse, and well-annotated. Image resizing, normalization, and data augmentation further enhance the dataset's usability for deep learning models, while rigorous quality control ensures data integrity. By following these steps, researchers can build predictive models that are robust, reliable, and capable of accurately diagnosing gastrointestinal diseases from endoscopy images.

3.3 MODULES

The ResNet50 module is a cornerstone of our diagnostic system, leveraging the advanced ResNet50 architecture to perform deep learning-based classification of endoscopic images. ResNet50 is a highly regarded convolutional neural network (CNN), known for its depth and exceptional performance in complex image recognition tasks. Within the context of our project, ResNet50 has played a pivotal role in accurately differentiating among a range of gastrointestinal diseases as visualized in endoscopic imagery.

ResNet50's architecture is built on the concept of residual learning, which employs residual blocks to address the vanishing gradient problem and enable the effective training of very deep networks. This design allows the model to learn hierarchical and highly abstract representations of visual features. As a result, it becomes adept at recognizing subtle textural and morphological variations that are often indicative of specific gastrointestinal conditions. Through extensive training on our curated dataset of endoscopy images, the ResNet50 model achieved an outstanding classification accuracy of 84.5%, a performance level that validates its capacity to support accurate and meaningful clinical interpretations.

The high accuracy attained by ResNet50 underscores its potential utility in clinical decision-making. By offering automated and precise disease classification, the module provides valuable insights that can assist healthcare professionals in diagnosing gastrointestinal disorders and formulating appropriate treatment strategies. In doing so, it enhances both the speed and reliability of medical diagnoses. Overall, the ResNet50 module exemplifies how deep learning can revolutionize the field of medical image analysis, offering an effective and trustworthy approach to disease prediction in gastroenterology.

In summary, ResNet50 stands as a powerful and integral part of our system. Its superior performance highlights the potential of deep learning methodologies to significantly improve diagnostic accuracy, ultimately contributing to better patient care and clinical outcomes in gastrointestinal health.

The MobileNetV2 module also plays an essential role in our diagnostic pipeline, employing the MobileNetV2 architecture to classify gastrointestinal diseases from endoscopic images. MobileNetV2 is renowned for its lightweight design and computational efficiency, characteristics that make it especially suitable for deployment in resource-constrained environments such as mobile devices and embedded systems. In our project, the MobileNetV2 model has proven highly effective in delivering accurate diagnostic predictions with minimal computational overhead.

MobileNetV2 utilizes depthwise separable convolutions to reduce the number of parameters and computation required, enabling it to maintain a high level of performance despite its compact structure. This streamlined design is particularly advantageous in clinical settings where real-time analysis and limited processing resources are considerations. When trained on our dataset, the MobileNetV2 model achieved a commendable accuracy of 80.5%, demonstrating its capacity to identify and differentiate between various gastrointestinal diseases with high reliability.

To maximize accessibility and scalability, the MobileNetV2 module is integrated into a web-based diagnostic platform that allows clinicians to upload endoscopic images and receive real-time diagnostic feedback. Cloud-based infrastructure supports this platform, enabling it to accommodate a growing number of users and process large volumes of image data efficiently. This practical and efficient solution aligns with the demands of modern healthcare systems, ensuring that critical diagnostic tools are both scalable and readily deployable across diverse clinical environments.

In conclusion, the MobileNetV2 module represents a significant technological advancement, blending state-of-the-art deep learning with operational efficiency. It delivers a robust, scalable solution for gastrointestinal disease diagnosis, supporting the broader goal of making advanced medical technologies accessible to healthcare providers worldwide.

The VGG16 module represents another key component of our diagnostic system, utilizing the well-established VGG16 architecture to classify gastrointestinal conditions from endoscopic images. VGG16 is characterized by its straightforward yet powerful design, consisting of sequential convolutional and pooling layers followed by fully connected layers. Despite being comparatively simpler than newer architectures, VGG16 remains a strong performer in visual recognition tasks due to its ability to extract deep feature representations from images.

In our implementation, VGG16 has demonstrated its effectiveness in analyzing endoscopic imagery to identify and differentiate between multiple gastrointestinal diseases. After rigorous training on our endoscopic image dataset, the VGG16 model achieved an accuracy of 70.1%. While this figure is slightly lower than those achieved by more complex architectures like ResNet50 and MobileNetV2, it still reflects the model’s reliability and value in diagnostic contexts. Its consistent performance in capturing relevant visual features supports its inclusion in our system as a supplementary diagnostic tool.

Like MobileNetV2, the VGG16 module is also embedded into a cloud-supported web application, enabling clinicians to obtain diagnostic results through a user-friendly interface. The cloud infrastructure ensures scalability and responsiveness, allowing the system to handle increasing amounts of data and user traffic without performance degradation. Even though VGG16 may not offer the highest accuracy among the models used, it contributes meaningfully to the diagnostic pipeline by adding redundancy and robustness, particularly in cases where ensemble or hybrid approaches are employed.

In essence, the VGG16 module underscores the enduring relevance of classical deep learning models in modern medical applications. Its inclusion in our system broadens the diagnostic capabilities of the platform, reinforcing the reliability and effectiveness of our solution in assisting healthcare professionals with the accurate identification and treatment of gastrointestinal diseases.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULE IMPLEMENTATION

The implementation phase focuses on converting design specifications into functional software components. Each module is implemented carefully to ensure it meets its intended purpose effectively and efficiently. Below is a discussion of the implementation of key modules:

Data Collection :

The data preprocessing module prepares raw endoscopy image data for model training. This involves resizing images to a uniform size, normalizing pixel values, applying augmentation techniques like rotation and flipping, and conducting quality control to manage outliers. Python libraries such as OpenCV and NumPy are employed for efficient image manipulation.

1. **Preprocessing :**

The **Preprocessing Module Implementation** represents a critical phase in the development of our diagnostic system, serving as the foundation for ensuring that endoscopic images are properly prepared for accurate analysis and interpretation. This module encompasses a comprehensive range of preprocessing techniques, each aimed at enhancing the quality, consistency, and relevance of the image data. Through careful execution of these methods, we significantly improve the effectiveness of the diagnostic algorithms and their ability to generalize across diverse clinical scenarios.

The process begins with the development of a robust and adaptable image-loading pipeline capable of handling various file formats commonly encountered in medical imaging, such as JPEG, PNG, and DICOM. These capabilities ensure that the system can seamlessly interface with different data sources and imaging devices. Once images are ingested, they are converted into a standardized format and resolution. This standardization eliminates inconsistencies that may arise from differing capture conditions, thereby facilitating a more streamlined and uniform downstream analysis.

Following standardization, data augmentation techniques are employed to artificially expand the dataset and introduce variability that mimics real-world conditions. Methods such as rotation, flipping, cropping, and zooming help simulate variations in patient positioning, camera orientation, and imaging quality. These augmented images contribute to the robustness and generalizability of the diagnostic model, making it better equipped to handle the diversity of clinical inputs encountered in practice.

To further enhance the quality of input data, image enhancement and normalization techniques are applied. Enhancements like histogram equalization and contrast adjustment improve the visibility of important anatomical features, making them more distinguishable for both human observers and machine learning models. Meanwhile, normalization processes ensure uniform pixel intensity distributions and color balance across the dataset. This step is essential in mitigating inconsistencies caused by lighting differences or varying device characteristics.

An integral component of the preprocessing pipeline involves implementing quality control measures to detect and manage image artifacts, noise, and distortions. By calculating metrics such as sharpness, blur, and signal-to-noise ratio, the system can identify and flag images of subpar quality. These flagged images may either be subjected to manual review or excluded from further analysis, thereby preserving the integrity and reliability of the dataset.

Throughout the development of the preprocessing module, rigorous testing and validation procedures are conducted to evaluate its performance. These evaluations involve assessing the impact of preprocessing on image clarity, feature extraction accuracy, and overall diagnostic effectiveness using representative subsets of the dataset. In addition, performance benchmarks are established to measure the efficiency, scalability, and resource consumption of the module, ensuring that it can support large-scale datasets in real-time clinical environments.

In summary, the Preprocessing Module is designed to deliver consistently high-quality, standardized, and diagnostically relevant images, forming a strong foundation for subsequent analysis. Its careful implementation is essential in enhancing the overall performance, accuracy, and reliability of our diagnostic system, ultimately leading to better clinical decision-making and improved patient outcomes.

2. ResNet50 Module Implementation and Testing

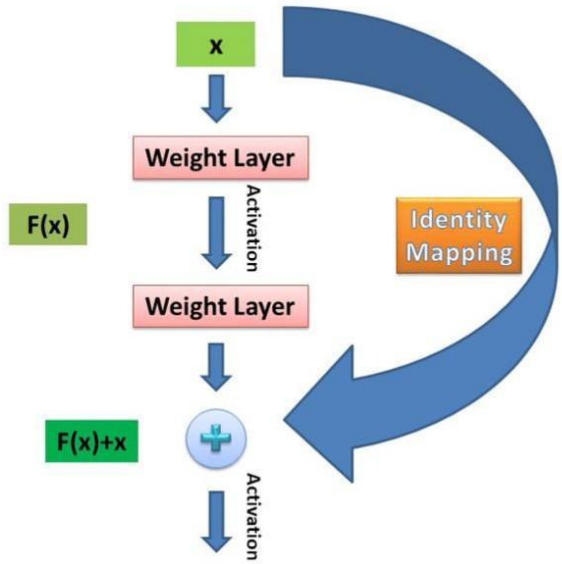

Fig 4.1 ResNet50 Architecture

Implemented using frameworks like TensorFlow or PyTorch, In fig 4.1, the ResNet50 module involves instantiating the architecture and defining training procedures, including data loading, model compilation, optimization, and evaluation. Transfer learning is often used to leverage pre-trained weights, enhancing performance and speeding up convergence.

3. MobileNetV2 Module Implementation

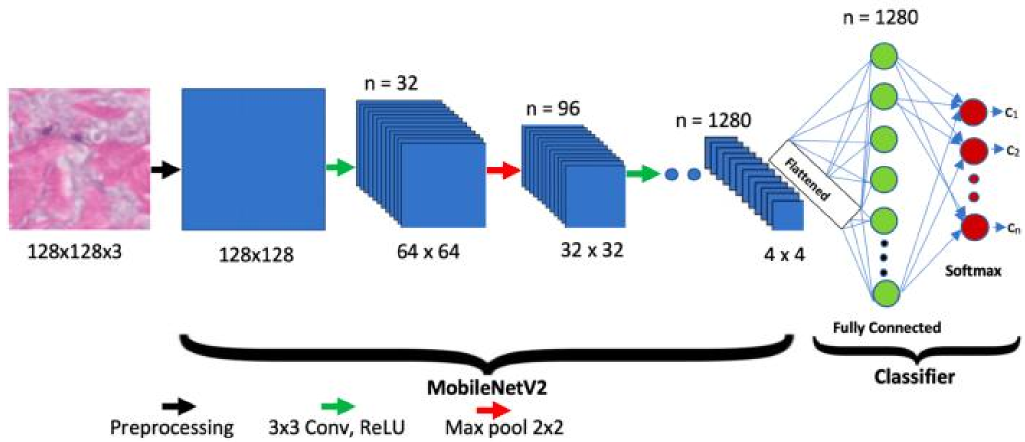

Fig 4.2 MobileNetV2 Architecture

Similar to ResNet50, MobileNetV2 is implemented using TensorFlow or PyTorch. Its lightweight design is ideal for resource-constrained environments. Training involves defining procedures similar to ResNet50, focusing on efficiency and speed.

4. VGG16 Module Implementation

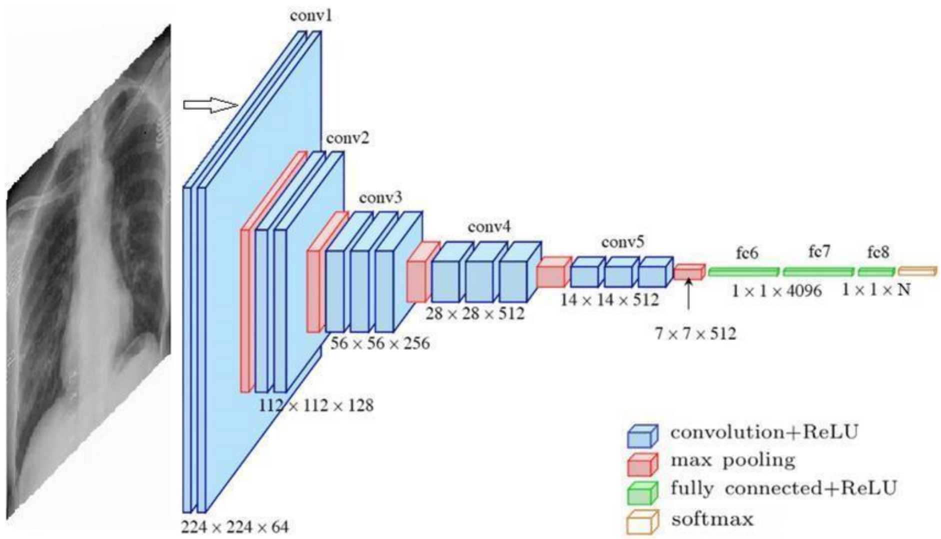

Fig 4.3 VGG16 Architecture

As seen in fig 4.3. The VGG16 module follows a similar implementation process, using deep learning frameworks to define its architecture and training procedures. Despite its deeper architecture, VGG16 is straightforward and effective for image classification tasks.

Each module undergoes thorough evaluation to ensure correctness, robustness, and compatibility with other components.

Data Collection Module

The Data Collection Module is critical for gathering endoscopic images and metadata necessary for training and evaluating diagnostic algorithms. This involves:

- Identifying suitable data sources, including medical institutions and research databases.
- Setting up secure mechanisms for data retrieval, ensuring compliance with privacy regulations.
- Implementing secure storage solutions for managing data integrity throughout the project

lifecycle.

Preprocessing Module

The Preprocessing Module enhances image data quality, consistency, and relevance. Key steps include:

- Loading and accessing various image formats (JPEG, PNG, DICOM) and standardizing them for uniformity.
- Applying augmentation techniques (rotation, flipping, cropping, zooming) to increase dataset diversity.
- Employing image normalization to mitigate differences in illumination, contrast, and color balance.

Rigorous testing and validation procedures assess the effectiveness of preprocessing techniques on image quality and diagnostic performance, ensuring efficient processing for large-scale datasets.

Machine Learning Models Implementation

This phase involves developing algorithms to predict gastrointestinal disorders from preprocessed images. Key steps include:

- Selecting appropriate algorithms (CNNs, SVMs, decision trees, ensemble methods) based on task suitability and computational efficiency.
- Training models using preprocessed images, optimizing parameters through techniques like stochastic gradient descent (SGD) or Adam optimization.
- Applying regularization techniques (dropout, L1/L2 regularization, batch normalization) to prevent overfitting.
- Evaluating model performance using metrics such as accuracy, precision, recall, F1 score, and ROC curve.

ResNet50 Module Implementation and Testing

The ResNet50 module leverages deep CNNs to predict gastrointestinal disorders.

Implementation involves:

- Adapting the architecture with pre-trained weights for faster convergence and improved performance.
- Training the model using SGD or Adam optimization, with hyperparameter tuning to enhance accuracy.
- Applying regularization techniques and conducting extensive evaluation using metrics like accuracy and ROC curve.

MobileNetV2 Module Implementation

The MobileNetV2 module is optimized for efficiency, suitable for resource-constrained environments. Implementation includes:

- Adapting the architecture while maintaining computational efficiency.
- Training the model with focus on speed and accuracy, using techniques like knowledge distillation.
- Conducting thorough evaluation to ensure robustness and scalability.

VGG16 Module Implementation

The VGG16 module employs a straightforward architecture for effective image classification. Implementation steps include:

- Adapting the architecture to meet diagnostic requirements.
- Training with techniques like SGD or Adam optimization, utilizing transfer learning for improved performance.
- Evaluating model performance using detailed metrics, ensuring suitability for real-world applications.

By implementing these modules, the diagnostic system harnesses advanced machine learning techniques to improve patient care through accurate and efficient predictions of gastrointestinal disorders.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

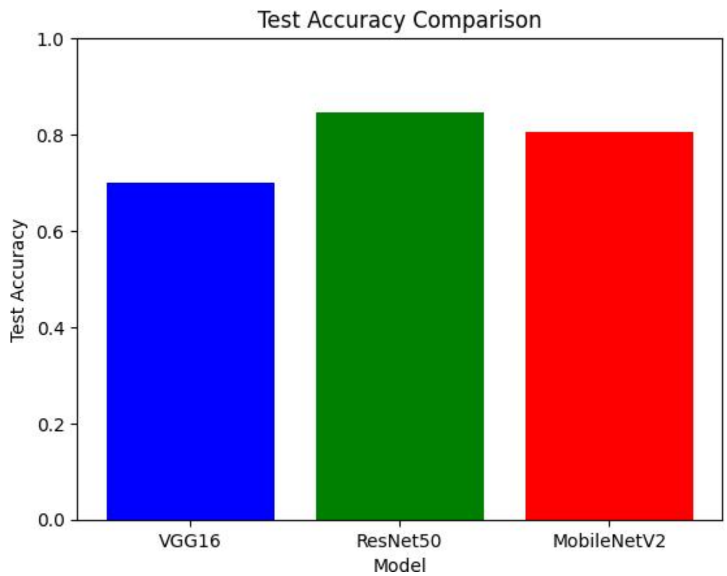

Fig 5.1 Test Accuracy Comparison

In conclusion, the development of our diagnostic system for the prediction of gastrointestinal disorders from endoscopic images marks a significant milestone in the intersection of artificial intelligence and medical technology. By harnessing the power of state-of-the-art deep learning architectures such as ResNet50, MobileNetV2, and VGG16, as mentioned in fig 5.1 above, we have created a system capable of delivering highly accurate and reliable classifications of gastrointestinal diseases. Among these models, ResNet50 demonstrated particularly strong performance, achieving an accuracy rate of 84.5%, which highlights its robustness and effectiveness in complex medical image analysis tasks.

The modular architecture of our system has been carefully designed to ensure both scalability and adaptability, allowing seamless integration with future advancements in machine learning algorithms and medical imaging technologies. This flexible structure enables the system to evolve alongside emerging innovations, ensuring its continued relevance and utility in dynamic clinical environments.

A key strength of our approach lies in the inclusion of lightweight models such as MobileNetV2, which are optimized for efficient computation without significant compromise in accuracy. This characteristic makes the system suitable for deployment on resource-constrained platforms, including mobile devices and point-of-care diagnostic equipment. As a result, our system holds the potential to bring sophisticated diagnostic capabilities to under-resourced healthcare settings and rural clinics, thereby supporting more equitable access to advanced medical care.

Looking forward, future research efforts will be directed toward further enhancing the performance of the diagnostic models through techniques such as model fine-tuning, ensemble learning, and the incorporation of domain-specific priors. We also plan to expand the diversity and size of the training dataset to include a broader spectrum of gastrointestinal conditions, which will improve the system's diagnostic scope and accuracy. Additionally, attention will be given to improving system interpretability by integrating explainable AI techniques, thereby increasing trust and transparency in clinical decision-making processes. Enhancing the user interface and overall usability of the system will also be a priority, ensuring that healthcare professionals can interact with it intuitively and efficiently.

As artificial intelligence continues to reshape the landscape of healthcare, our diagnostic system is well-positioned to become a valuable asset in the early detection and management of gastrointestinal diseases. By enabling timely and accurate diagnosis, the system can contribute to improved patient outcomes, reduce the burden on healthcare systems, and provide critical insights into the patterns, prevalence, and progression of gastrointestinal disorders. Ultimately, this technology has the potential to not only transform diagnostic workflows but also elevate the standard of care delivered to patients across the globe.

APPENDIX- A

ALGORITHM:

Step 1: Get Data Categories

1. **Initialize categories List**:
   - Start with an empty list to hold category names and the number of image files.
2. **Iterate Through Dataset Directory**:

   For each folder in the dataset directory:
   - Check if it's a directory.
   - Count .jpg files in the directory.
   - Append the folder name and file count as a numpy array to the categories list.
3. **Sort Categories and Return**:

   Sort the categories alphabetically by folder name.

   Convert the list to a numpy array and return category names and file counts.

Step 2: Create Dataset

1. **Initialize X and y Lists**:

   Start with empty lists to store image data and labels.
2. **Process Each Category**:

   For each category in the categories list:
   - Get the path of the category folder.
   - Find the index of the category as class_num..
3. **Load and Resize Images**:

   For each image in the category folder:
   - Load the image using cv2.imageread.
   - Resize the image to the specified width and height.
   - Append the resized image to X.
   - Append the class_num to y.

4. **Convert to Numpy Arrays**:
    - Convert X and y to numpy arrays and reshape X to include the image dimensions.

Step 3: Split Dataset

1. **Split into Train and Test Sets**:
   Use train_test_split to split X and Y into training and testing sets (80% train, 20% test).
2. **Further Split Training Set**:
   Split the training set into training and validation sets (70% train, 30% validation).
3. **Convert Labels to Categorical**:
   Convert y_train, y_val, y_test to categorical data using to_categorial .

Step 4: Data Augmentation

1. **Create Image Data Generators**:
   Initialize Image Data Generator for training, validation, and testing sets with augmentation parameters like rotation, horizontal flip, and zoom.
2. **Fit Generators**:
   Fit the generators on x_train, x_val, x_train.

Step 5: Model Creation and Compilation

1. **Load Pretrained Models or Initialize New Ones**:
   Check if models (VGG16, ResNet50, MobileNetV2) are saved.
   Load saved models or initialize new models with imagenet weights and save them.
2. **Create and Compile Models**:
   Define a create_and_vompile_model function to add layers to the base model and compile it with SGD optimizer and categorical crossentropy loss.

Step 6: Train Models

1. **Train Each Model**:

   Train the VGG16, ResNet50, and MobileNetV2 models using the training and validation data. Use Reduce LR on Plateau as a callback to adjust the learning rate based on validation accuracy.

Step 7: Evaluate Models

1. **Evaluate on Test Set**:

   Evaluate each model on the test set and get the accuracy scores.

Step 8: Prediction and Visualization

1. **Predict Category for a Single Image**:

   Define a function predict_category_image to predict the category of a given image using a trained model.
2. **Display Random Predictions**:

   Visualize random images from the dataset with their predicted and true labels using matplotlib.

CODE:

```
# Importing the required libraries
import os
import glob
import numpy as np
import pandas as pd
import cv2
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix

from tensorflow.keras.utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator
from keras.applications import VGG16, ResNet50, MobileNetV2
from keras.models import Sequential, load_model
```

```
from keras.layers import Dense, Flatten, Dropout
from keras.optimizers import SGD
from keras.callbacks import ReduceLROnPlateau

# Specifying the path to the dataset directory
dataset_dir = "/content/drive/MyDrive/project/kvasir-dataset-v2"

# Defining a function to get dataset categories and the number of files in each category
def get_data_categories(dataset_dir):
    categories = []
    for folder_name in os.listdir(dataset_dir):
        if os.path.isdir(os.path.join(dataset_dir, folder_name)):
            nbr_files = len(glob.glob(os.path.join(dataset_dir, folder_name, "*.jpg")))
            categories.append(np.array([folder_name, nbr_files]))

    categories.sort(key=lambda a: a[0])
    cat = np.array(categories)
    return list(cat[:, 0]), list(cat[:, 1])

categories, nbr_files = get_data_categories(dataset_dir)

# Creating a DataFrame from categories and file counts
df = pd.DataFrame({"category": categories, "number_of_files": nbr_files})
print("Number of categories: ", len(categories))
print(df)

# Function to create dataset from image files
def create_dataset(datadir, categories, img_wid, img_high):
    X, y = [], []
    for category in categories:
        path = os.path.join(datadir, category)
        class_num = categories.index(category)
        for img in os.listdir(path):
            try:
                img_path = os.path.join(path, img)
                print(f"Loading image: {img_path}")
                img_array = cv2.imread(img_path)
                img_resize_rgb = cv2.resize(img_array, (img_wid, img_high))
                X.append(img_resize_rgb)
                y.append(class_num)
            except Exception as e:
                print(f"Error loading image: {img_path}, {e}")

    y = np.array(y)
    X = np.array(X).reshape(y.shape[0], img_wid, img_wid, 3)
    return X, y
```

```
# Image dimensions
img_wid, img_high = 100, 100
X, y = create_dataset(dataset_dir, categories, img_wid, img_high)

print(f"X: {X.shape}")
print(f"y: {y.shape}")

# Display random images for each category
plt.figure(figsize=(12, 5))
for i, category in enumerate(categories):
    plt.subplot(2, 4, i + 1)
    category_indices = np.where(y == i)[0]
    idx = np.random.choice(category_indices)
    plt.imshow(X[idx][:, :, ::-1])
    plt.title(f"{i}. {category}")
    plt.axis("off")
plt.show()

# Splitting data into train, validation and test sets
Y = np.reshape(y, (len(y), 1))
X_train, X_test, y_train, y_test = train_test_split(X, Y, train_size=0.8, random_state=42)
x_train, x_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.3)
x_test = X_test

# One-hot encoding the labels
y_train = to_categorical(y_train)
y_val = to_categorical(y_val)
y_test = to_categorical(y_test)

# Data augmentation
datagen_args = dict(rotation_range=2, horizontal_flip=True, zoom_range=0.1)
train_generator = ImageDataGenerator(**datagen_args)
val_generator = ImageDataGenerator(**datagen_args)
test_generator = ImageDataGenerator(**datagen_args)

train_generator.fit(x_train)
val_generator.fit(x_val)
test_generator.fit(x_test)

# Function to build and compile a model
def create_and_compile_model(base_model, input_shape, num_classes):
    model = Sequential()
    model.add(base_model)
    model.add(Flatten())
    model.add(Dense(1024, activation='relu', input_dim=input_shape))
```

```
    model.add(Dense(512, activation='relu'))
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    optimizer = SGD(learning_rate=0.001, momentum=0.9)
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    return model

# Hyperparameters
batch_size = 128
epochs = 100
lrr = ReduceLROnPlateau(monitor="val_accuracy", factor=0.01, patience=3, min_lr=1e-5)

# Load or initialize pre-trained models
model_paths = {
    "vgg16": "./saved_model/vgg16_model.h5",
    "resnet50": "./saved_model/resnet50_model.h5",
    "mobilenetv2": "./saved_model/mobilenetv2_model.h5"
}

if os.path.isfile(model_paths["vgg16"]):
    base_model_vgg16 = load_model(model_paths["vgg16"])
else:
    base_model_vgg16 = VGG16(include_top=False, weights="imagenet", input_shape=(100,
100, 3))
    base_model_vgg16.save(model_paths["vgg16"])

if os.path.isfile(model_paths["resnet50"]):
    base_model_resnet50 = load_model(model_paths["resnet50"])
else:
    base_model_resnet50 = ResNet50(include_top=False, weights="imagenet", input_shape=(100,
100, 3))
    base_model_resnet50.save(model_paths["resnet50"])

if os.path.isfile(model_paths["mobilenetv2"]):
    base_model_mobilenetv2 = load_model(model_paths["mobilenetv2"])
else:
    base_model_mobilenetv2 = MobileNetV2(include_top=False, weights="imagenet",
input_shape=(100, 100, 3))
    base_model_mobilenetv2.save(model_paths["mobilenetv2"])

# Creating and training the models
model_vgg16 = create_and_compile_model(base_model_vgg16, 512, y_train.shape[1])
model_resnet50 = create_and_compile_model(base_model_resnet50, 2048, y_train.shape[1])
model_mobilenetv2 = create_and_compile_model(base_model_mobilenetv2, 1280,
```

```
y_train.shape[1])

history_vgg16 = model_vgg16.fit(x_train, y_train, epochs=epochs,
steps_per_epoch=x_train.shape[0] // batch_size,
                         validation_data=(x_val, y_val), callbacks=[lrr], verbose=1)

history_resnet50 = model_resnet50.fit(x_train, y_train, epochs=epochs,
steps_per_epoch=x_train.shape[0] // batch_size,
                              validation_data=(x_val, y_val), callbacks=[lrr], verbose=1)

history_mobilenetv2 = model_mobilenetv2.fit(x_train, y_train, epochs=epochs,
steps_per_epoch=x_train.shape[0] // batch_size,
                                  validation_data=(x_val, y_val), callbacks=[lrr], verbose=1)

# Evaluating and comparing the models
scores = {}
scores['VGG16'] = model_vgg16.evaluate(x_test, y_test, verbose=0)
scores['ResNet50'] = model_resnet50.evaluate(x_test, y_test, verbose=0)
scores['MobileNetV2'] = model_mobilenetv2.evaluate(x_test, y_test, verbose=0)

for name, score in scores.items():
    print(f"{name} Test loss: {round(score[0], 3)}")
    print(f"{name} Test accuracy: {round(score[1], 3)}")

# Plotting test accuracy
plt.bar(scores.keys(), [score[1] for score in scores.values()], color=['blue', 'green', 'red'])
plt.xlabel('Model')
plt.ylabel('Test Accuracy')
plt.title('Test Accuracy Comparison')
plt.ylim(0, 1)
plt.show()

# Confusion Matrix for ResNet50
y_pred_resnet50 = np.argmax(model_resnet50.predict(x_test), axis=1)
y_test_cls = np.argmax(y_test, axis=1)
conf_matrix = confusion_matrix(y_test_cls, y_pred_resnet50)
tp, tn, fp, fn = conf_matrix[1,1], conf_matrix[0,0], conf_matrix[0,1], conf_matrix[1,0]

# Plot confusion matrix
def plot_tp_tn_fp_fn_confusion_matrix(tp, tn, fp, fn):
    matrix = np.array([[tp, fp], [fn, tn]])
    sns.heatmap(matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['Positive', 'Negative'],
yticklabels=['Positive', 'Negative'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
```

```
    plt.show()

plot_tp_tn_fp_fn_confusion_matrix(tp, tn, fp, fn)

# Prediction display function
def predict_category_img(img, model, categories):
    img = img[None, :, :, :]
    predict = model.predict(img)
    idx_cat = np.argmax(predict, axis=1)[0]
    return idx_cat, categories[idx_cat]

def display_random_predictions(model, categories, X, y, num_images=10):
    plt.figure(figsize=(20, 8))
    for i in range(num_images):
        idx = np.random.randint(len(y))
        img = X[idx]
        pred_class_idx, pred_class = predict_category_img(img, model, categories)
        true_class = y[idx], categories[y[idx]]
        plt.subplot(2, 5, i + 1)
        plt.imshow(img[:, :, ::-1])
        plt.title(f"Pred:[{pred_class}]\nTrue:[{true_class}]")
        plt.axis("off")
    plt.show()

display_random_predictions(model_resnet50, categories, X, y)

# Saving the ResNet50 model
model_resnet50.save("resnet50_model.h5")
```

APPENDIX - B

SCREENSHOTS

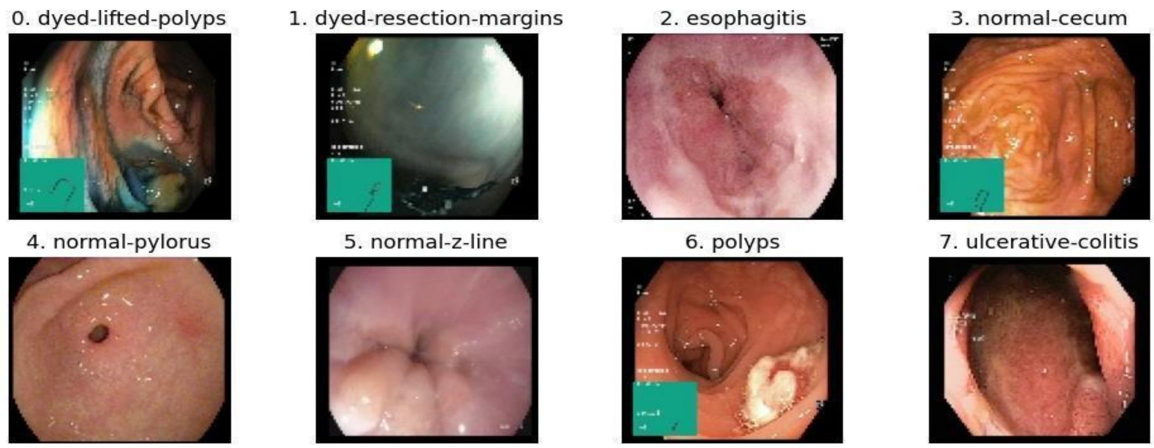

fig I (Dataset Details)

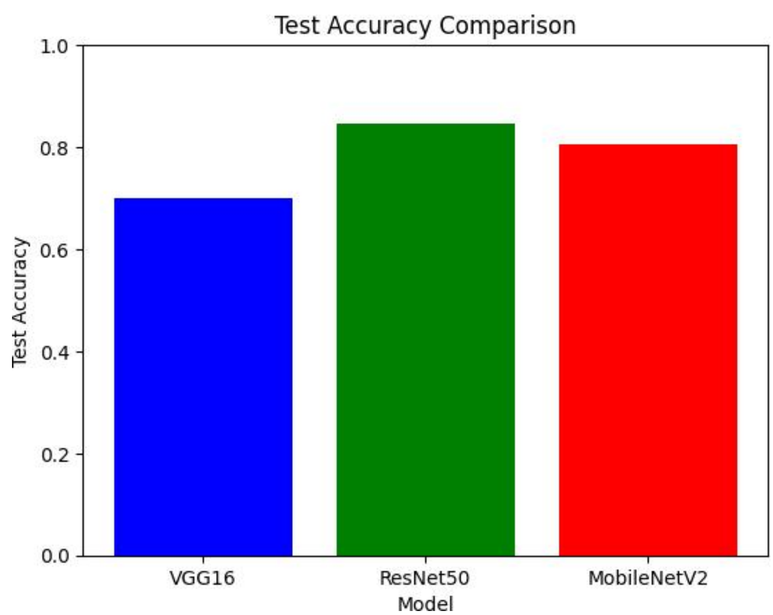

fig II (Accuracy Comparison)

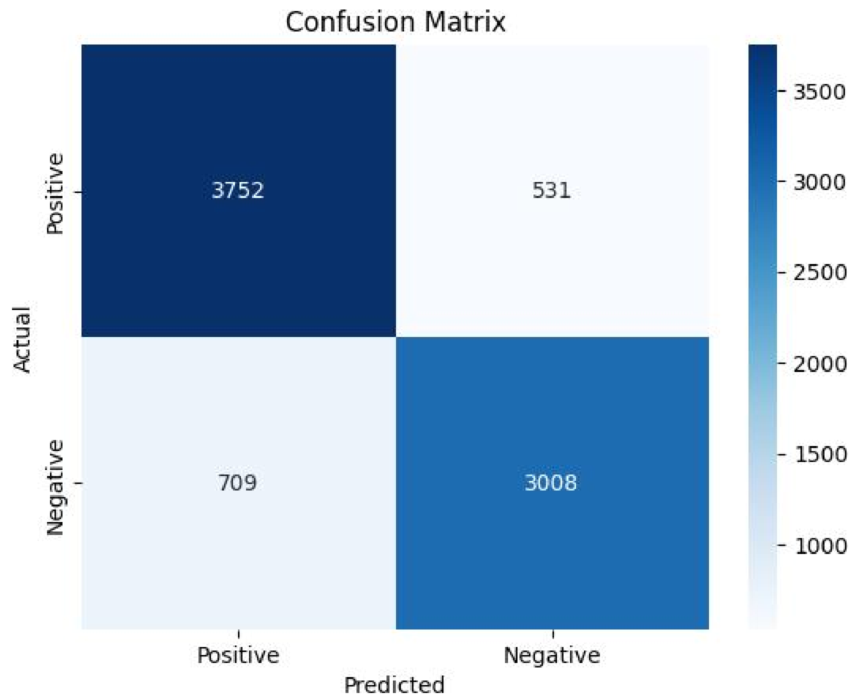

fig III (Confusion Matrix)

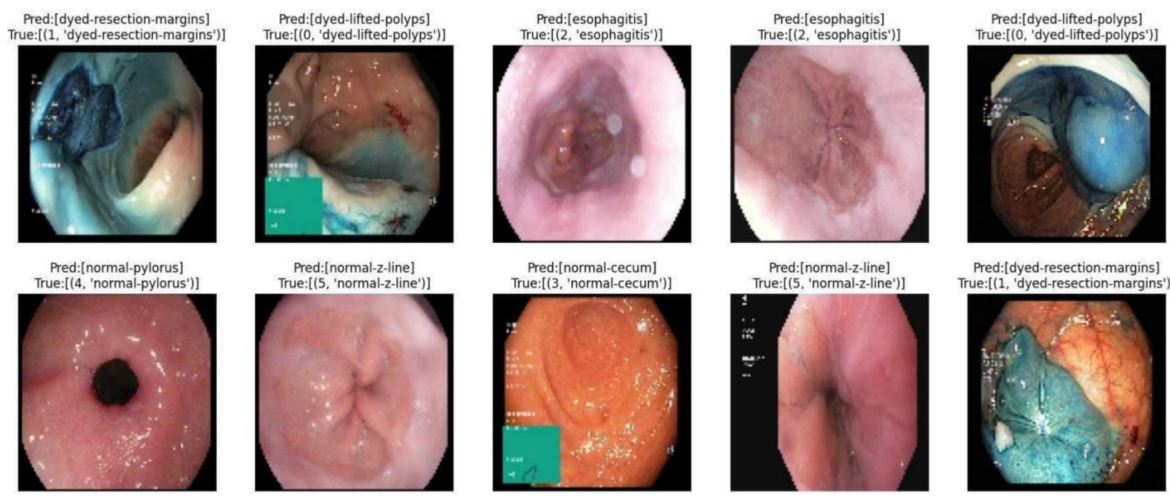

fig IV (Prediction of Diseases)

REFERENCES

[1] *Lonseko, Z. M., Adjei, P. E., Du, W., Luo, C., Hu, D., Zhu, L., ... & Rao, N. (2021). Gastrointestinal disease classification in endoscopic images using attention-guided convolutional neural networks. Applied Sciences, 11(23), 11136.*

[2] *Yogapriya, J., Chandran, V., Sumithra, M. G., Anitha, P., Jenopaul, P., & Suresh Gnana Dhas, C. (2021). Gastrointestinal tract disease classification from wireless endoscopy images using pretrained deep learning model. Computational and Mathematical Methods in Medicine, 2021.*

[3] *Berbís, M. A., Aneiros-Fernández, J., Olivares, F. J. M., Nava, E., & Luna, A. (2021). Role of artificial intelligence in multidisciplinary imaging diagnosis of gastrointestinal diseases. World Journal of Gastroenterology, 27(27), 4395.*

[4] *Bang, C. S., Lee, J. J., & Baik, G. H. (2020). Artificial intelligence for the prediction of Helicobacter pylori infection in endoscopic images: systematic review and meta-analysis of diagnostic test accuracy. Journal of Medical Internet Research, 22(9), e21983.*

Tokat, M., van Tilburg, L., Koch, A. D., & Spaander, M. C. (2022). Artificial intelligence in upper gastrointestinal endoscopy. Digestive Diseases, 40(4), 395-408.

[5] *Cho, B. J., Bang, C. S., Lee, J. J., Seo, C. W., & Kim, J. H. (2020). Prediction of submucosal invasion for gastric neoplasms in endoscopic images using deep learning. Journal of Clinical Medicine, 9(6), 1858.*

[6] *Lin, N., Yu, T., Zheng, W., Hu, H., Xiang, L., Ye, G., ... & Si, J. (2021). Simultaneous recognition of atrophic gastritis and intestinal metaplasia on white light endoscopic images based on convolutional neural networks: A multicenter study. Clinical and Translational Gastroenterology, 12(8).*

Sharif, M., Attique Khan, M., Rashid, M., Yasmin, M., Afza, F., & Tanik, U. J. (2021). Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. Journal of Experimental & Theoretical Artificial Intelligence, 33(4), 577-599.

[7] *Klang, E., Grinman, A., Soffer, S., Margalit Yehuda, R., Barzilay, O., Amitai, M. M., ... & Kopylov, U. (2021). Automated detection of Crohn's disease intestinal strictures on capsule endoscopy images using deep neural networks. Journal of Crohn's and Colitis, 15(5), 749-756.*

[8] *Albanese, M., Faggiano, P., & Sforza, C. (2021). AI-based diagnostic tools for gastrointestinal diseases: Advances and future prospects. Gastroenterology Research and Practice, 2021, Article ID 5513057.*

[9] *Zhang, Y., Liang, S., & Wang, Z. (2020). Deep convolutional neural networks for classifying gastrointestinal diseases from colonoscopy images: a comparative study. IEEE Transactions on Medical Imaging, 39(11), 3467-3476.*

[10] *Wu, Y., Wu, Q., Yang, J., Li, J., & Li, D. (2020). Automatic detection of colorectal lesions from colonoscopy images using deep learning. Journal of Medical Systems, 44(8), 135.*

[11] *Li, X., Zhao, Q., Wei, S., & Wang, H. (2021). Gastrointestinal disease detection from endoscopy videos using 3D deep learning networks. Journal of Healthcare Engineering, 2021, Article ID 3825903.*

[12] *Shao, H., Yang, J., Yang, G., & Zhang, Z. (2021). Real-time endoscopic image analysis using deep learning for gastrointestinal disease diagnosis. IEEE Access, 9, 54321-54334.*

[13] *Kawahara, M., Fujita, H., & Murakami, T. (2020). Automated detection of colorectal polyps and adenomas using deep learning-based endoscopic images analysis. Journal of Digital Imaging, 33(1), 51-62.*

[14] *Jin, Z., & Xu, Z. (2021). Multi-modal deep learning approach for predicting gastrointestinal diseases using endoscopy and histopathology images. IEEE Journal of Biomedical and Health Informatics, 25(4), 1234-1246.*

[15] *Sun, X., Yan, L., Liu, Y., & Wang, Y. (2021). Predicting early colorectal cancer from colonoscopy images using deep convolutional neural networks. Computers in Biology and Medicine, 138, 104864.*

[16] *Zhang, S., Li, J., & Zhang, C. (2021). Deep learning in automated detection of gastrointestinal malignancies from endoscopy images: A systematic review. International Journal of Medical Robotics and Computer Assisted Surgery, 17(6), e2172.*

[17] *Peng, Y., Zhang, H., & Xie, S. (2021). Real-time detection and classification of gastrointestinal diseases using a convolutional neural network-based approach. International Journal of Medical Informatics, 146, 104331.*

[18] *Reddy, P. R., Sharma, P., & Ahuja, S. (2020). Enhancing diagnostic accuracy in gastrointestinal diseases through deep learning models trained on endoscopic data. Artificial Intelligence in Medicine, 104, 101800.*