Gastrointestinal Disease Detection

Group Members:

Aditya Ved (16010121208)
Anirudha Ta (16010121199)

Project Mentor:

Zaheed Shaikh

Abstract:

Every year many patients lose their lives due to cancer or infections in the intestinal tracts. The detection of polyps, which are generally harmless but some may become cancerous, is a very subjective matter and differs from doctor to doctor. An automated method to detect these polyps would provide a standard method for all doctors resulting in saving patients life in some cases.

As important as it is for the doctors to remove the polyps and cancerous growth in the intestine, another important thing is to make sure that the polyps are removed properly. An automated method to detect whether the dyed polyps have been removed properly or not is also required.

Thus in order to tackle all these problems, the proposed system aims to provide a reliable and efficient deep learning model coupled with a website for doctors to collaborate with each other and share their research works

List of Figures :

List of Tables :

Nomenclature :

ML - Machine Learning

CNN - Convolutional Neural Network

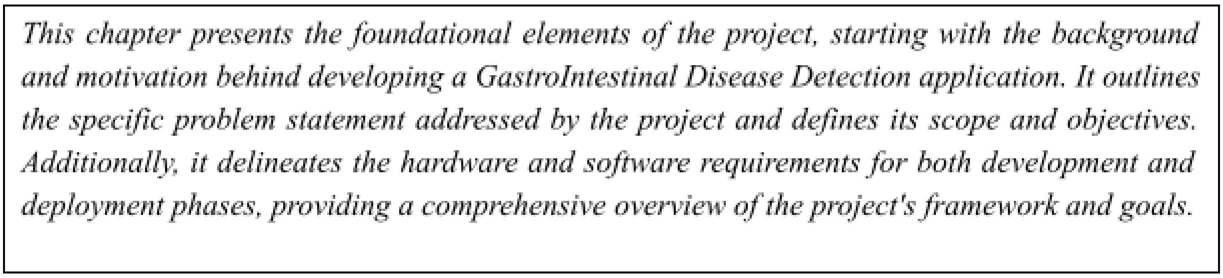
VGG - Visual Geometry Group

GI - Gastrointestinal

ReLU - Rectified Linear Unit

Chapter 1

Introduction



This chapter presents the foundational elements of the project, starting with the background and motivation behind developing a GastroIntestinal Disease Detection application. It outlines the specific problem statement addressed by the project and defines its scope and objectives. Additionally, it delineates the hardware and software requirements for both development and deployment phases, providing a comprehensive overview of the project's framework and goals.

1.1 Background / Motivation

1) There are many types of GI diseases, some of them are cancerous while others are non cancerous. Automatic detection of GI diseases along with automated report generation can prove to be extremely useful for doctors

2) For colorectal cancer prevention, endoscopic detection and removal of possible precancerous lesions are essential. Adenoma detection is therefore considered to be an important quality indicator in colorectal cancer screening. However, the ability to detect adenomas varies between doctors. Here a ML model will help in accurate detection of adenoma and other types of tumors that may go unnoticed by doctors

3) Several studies have shown that polyps are often overlooked during colonoscopies, with polyp miss rates of 14%-30% depending on the type and size of the polyps. Increasing the detection of polyps has been shown to decrease risk of colorectal cancer.

All the above reasons have motivated the project with the vision to provide doctors with quality second opinion to aid them give better diagnosis to the patients

1.2 Problem Statement

To perform Gastrointestinal Disease Detection of the available images of operations of patients and provide a platform (to admins and doctors) to facilitate this process for the hospital's internal system.

1.3 Scope

Our project’s aim is to detect Gastrointestinal Disease from the image provided. We have used many deep learning models to train the final model to detect Gastrointestinal Disease with high accuracy.

We have created an inter-hospital portal for doctors and admins where the admins can create patients and control which doctors have access to their profile and the doctors can add the patients of authorized patients and get response from the model as a side helper. Moreover we have also created a chat application for doctors to interact with one another regarding patient’s cases on a secure platform. This allows integrity of data to be within hospital systems and is also useful for record purposes.

1.4 Objectives

To provide a platform for doctors and administrators of the hospital for the below:

Doctor:

1. Ensure secure interaction between doctors of the hospital on web app’s inbuilt chat application to ensure confidentiality of patient’s medical information.
2. Easily detect the Gastrointestinal Disease by uploading the image

Admin:

1. Ensure confidentiality of patient’s operation images by strictly restricting it’s view to authorized doctors only
2. Maintaining a list of patients of the hospital.

Deep Learning Model :

1. Develop a robust and efficient deep learning model capable of detecting GI Diseases with the ability to extract deep features from the endoscopic images

1.5 Hardware and software requirements for development

Hardware Requirements:

1. Operating System: Windows, macOS, or Linux
2. Processor: Intel Core i3 / AMD Ryzen 3 or higher (recommended)
3. RAM: 8GB or higher (recommended)
4. Storage: 1GB of available disk space (excluding the space required for IDEs and other tools)
5. Graphics: GPU capable of running the development environment smoothly (integrated graphics are generally sufficient)
6. Display: Minimum of 1366 x 768 resolution

Software Requirements:

1. Front End: HTML, CSS, JavaScript, jQuery
2. Back End: Fast API
3. Database: MongoDB
4. ML Framework: Tensorflow
5. Machine Learning Model: VGG19
6. Cryptography: Python's "cryptography" module
7. IDE: Visual Studio Code

1.6 Hardware and software requirements for deployment

1. Server: Linux Server
2. Storage: 1 GB Storage
3. Graphics : Integrated or No GPU (model can run on CPU but if GPU (GTX series) the prediction is faster)
4. Support for Python 3.11 +

Software Requirements:

1. Front End: HTML, CSS, JavaScript, jQuery
2. Back End: Fast API
3. Database: MongoDB
4. ML Framework: Tensorflow
5. Machine Learning Model: VGG19
6. Cryptography: Python's "cryptography" module

Chapter 2

Literature Survey

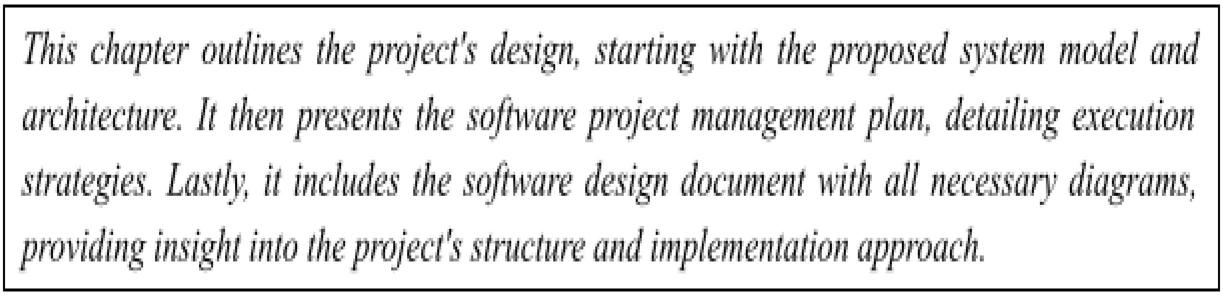
This chapter presents a review of research papers read and referred while building the proposed system. Various Deep Learning Architectures have been explored and their features have been listed in the table

Paper	Dataset Used	Title	Features	Accuracy
Segu et al.,	Kvasir		CNN-based method for extracting deep features, which resulted in superior classification performance when compared to the other handcrafted feature-based methods. considered a limited number of classes	96%
Gamage et al.,	Kvasir		used pre-trained DenseNet-201, ResNet-18, and VGG-16 CNN models as feature extractors with a global average pooling (GAP) layer to produce an ensemble of deep features as a single feature vector	97.38%
Khan et al.,	Kvasir		The ulcer annotated images are utilized to train the Mask RCNN model to obtain output in the form of	99.13 %

			bounding box ulcer detected area and mask segmented region. In the classification phase, the ResNet101 pre-trained CNN model is fine-tuned through transfer learning to derive deep features.	
Owais et al.,	Video Feeds		proposed a CNN and TML classification framework for the classification of several gastrointestinal diseases using endoscopic videos with a dataset containing a total of 77 video files with 52,471 images. They considered a total of 37 different classes (both diseased and normal cases) related to the human GI tract	92.7%

Chapter 3

Project Design



This chapter outlines the project's design, starting with the proposed system model and architecture. It then presents the software project management plan, detailing execution strategies. Lastly, it includes the software design document with all necessary diagrams, providing insight into the project's structure and implementation approach.

Chapter subtopics should be font 14 Times New Roman and 1.5 spacing bold, left aligned and justified

3.1 Proposed System model / Architecture

System Deployment Architecture follows the below patterns:

1. Microservices Architecture
   a. Client facing server
   b. Rest api server
   c. Authentication system microservice
   d. Machine Learning Model Server

2. Client Side Rendering Architecture

3. Asymmetric Key encryption to sign and verify JWTs (Json Web Tokens)

4. Asynchronous integration of machine learning model with WebApp.

The System architecture or the flow diagram of how microservices interact with each other in the Web Application can be viewed in the following image:

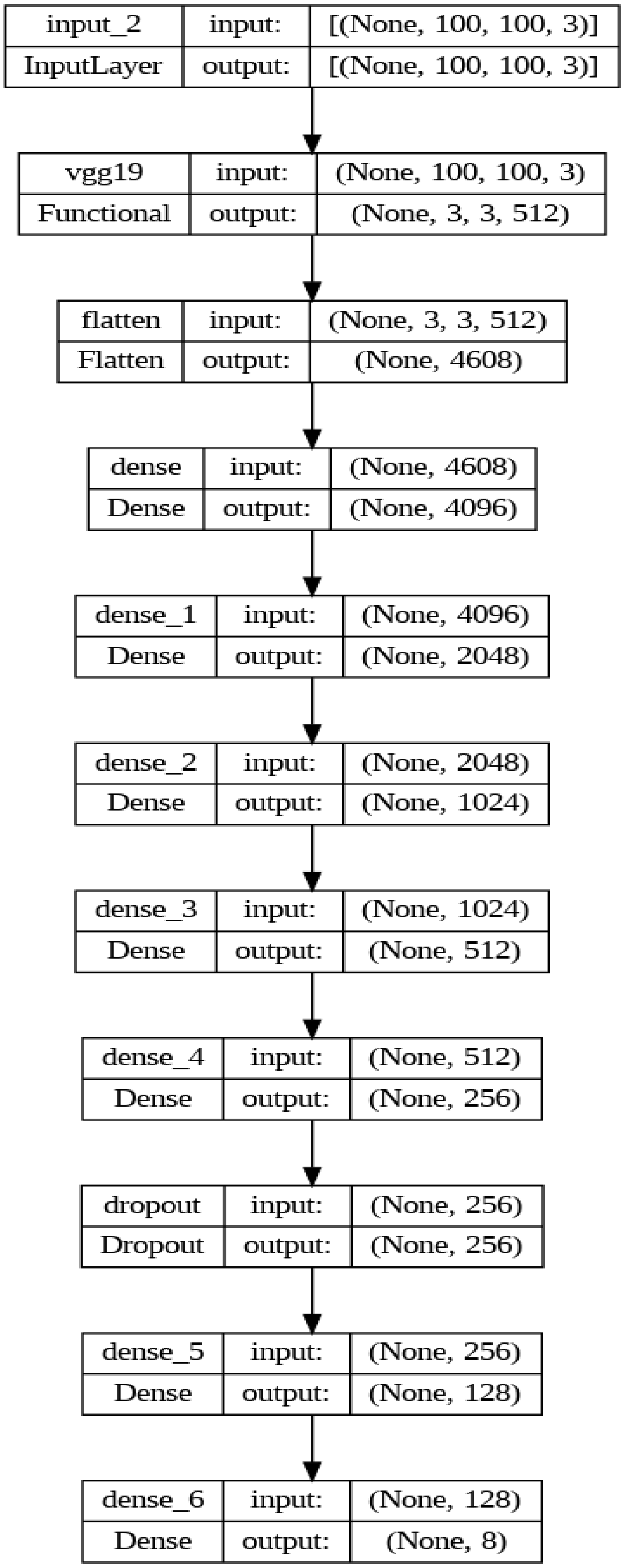
Figure 1 : **new_architecture.png**

(Kindly download the image to view it correctly)

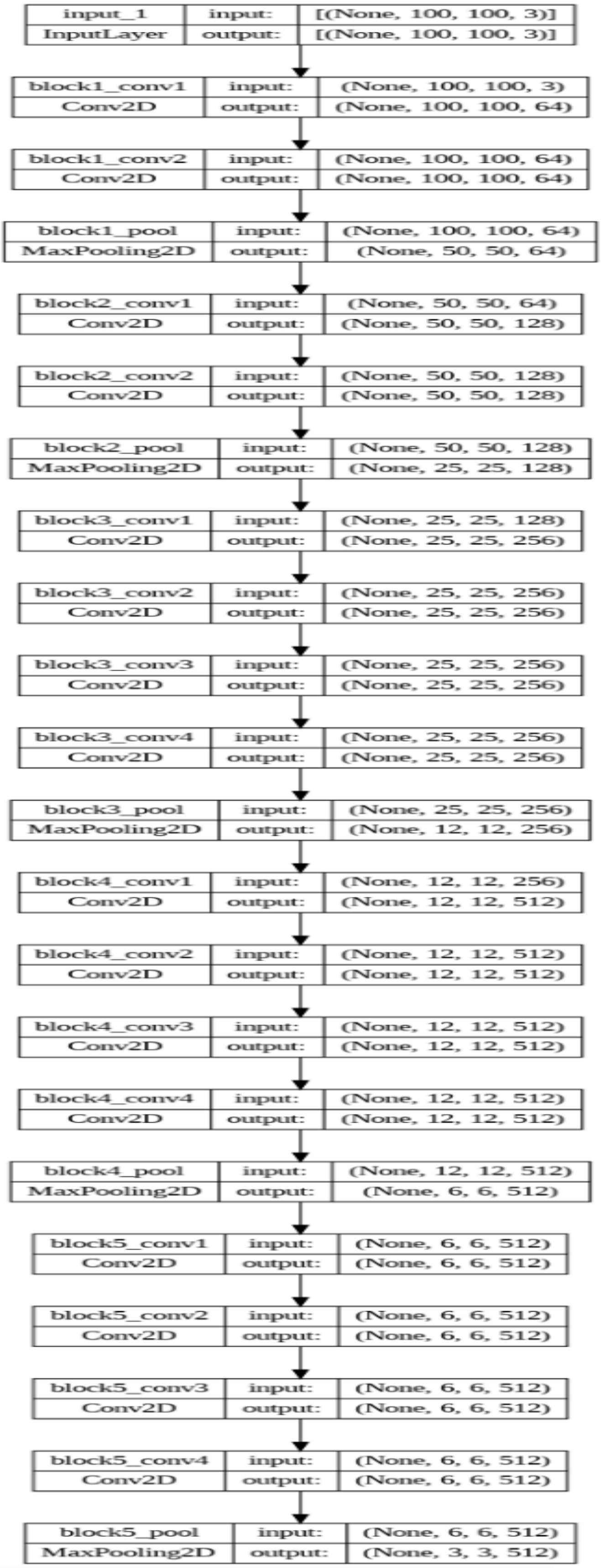
Model Architecture:

The Model Diagram to the right is the base VGG19 model and the diagram to the left is the custom VGG19 Model

Figure 2:



input_2	input:	[(None, 100, 100, 3)]
InputLayer	output:	[(None, 100, 100, 3)]
vgg19	input:	(None, 100, 100, 3)
Functional	output:	(None, 3, 3, 512)
flatten	input:	(None, 3, 3, 512)
Flatten	output:	(None, 4608)
dense	input:	(None, 4608)
Dense	output:	(None, 4096)
dense_1	input:	(None, 4096)
Dense	output:	(None, 2048)
dense_2	input:	(None, 2048)
Dense	output:	(None, 1024)
dense_3	input:	(None, 1024)
Dense	output:	(None, 512)
dense_4	input:	(None, 512)
Dense	output:	(None, 256)
dropout	input:	(None, 256)
Dropout	output:	(None, 256)
dense_5	input:	(None, 256)
Dense	output:	(None, 128)
dense_6	input:	(None, 128)
Dense	output:	(None, 8)



input_1	input:	[(None, 100, 100, 3)]
InputLayer	output:	[(None, 100, 100, 3)]
block1_conv1	input:	(None, 100, 100, 3)
Conv2D	output:	(None, 100, 100, 64)
block1_conv2	input:	(None, 100, 100, 64)
Conv2D	output:	(None, 100, 100, 64)
block1_pool	input:	(None, 100, 100, 64)
MaxPooling2D	output:	(None, 50, 50, 64)
block2_conv1	input:	(None, 50, 50, 64)
Conv2D	output:	(None, 50, 50, 128)
block2_conv2	input:	(None, 50, 50, 128)
Conv2D	output:	(None, 50, 50, 128)
block2_pool	input:	(None, 50, 50, 128)
MaxPooling2D	output:	(None, 25, 25, 128)
block3_conv1	input:	(None, 25, 25, 128)
Conv2D	output:	(None, 25, 25, 256)
block3_conv2	input:	(None, 25, 25, 256)
Conv2D	output:	(None, 25, 25, 256)
block3_conv3	input:	(None, 25, 25, 256)
Conv2D	output:	(None, 25, 25, 256)
block3_conv4	input:	(None, 25, 25, 256)
Conv2D	output:	(None, 25, 25, 256)
block3_pool	input:	(None, 25, 25, 256)
MaxPooling2D	output:	(None, 12, 12, 256)
block4_conv1	input:	(None, 12, 12, 256)
Conv2D	output:	(None, 12, 12, 512)
block4_conv2	input:	(None, 12, 12, 512)
Conv2D	output:	(None, 12, 12, 512)
block4_conv3	input:	(None, 12, 12, 512)
Conv2D	output:	(None, 12, 12, 512)
block4_conv4	input:	(None, 12, 12, 512)
Conv2D	output:	(None, 12, 12, 512)
block4_pool	input:	(None, 12, 12, 512)
MaxPooling2D	output:	(None, 6, 6, 512)
block5_conv1	input:	(None, 6, 6, 512)
Conv2D	output:	(None, 6, 6, 512)
block5_conv2	input:	(None, 6, 6, 512)
Conv2D	output:	(None, 6, 6, 512)
block5_conv3	input:	(None, 6, 6, 512)
Conv2D	output:	(None, 6, 6, 512)
block5_conv4	input:	(None, 6, 6, 512)
Conv2D	output:	(None, 6, 6, 512)
block5_pool	input:	(None, 6, 6, 512)
MaxPooling2D	output:	(None, 3, 3, 512)

3.2 Software Project Management Plan

Planning is very essential for successful completion of any activity in which multiple stakeholders are involved. To start with one will write down all activities needed to be carried out mentioning the role and responsibility of each human resource. This will also help in sequencing and tracking the progress of the development process. A sample Role and Responsibility matrix could be as follows, Please prepare according to the needs of your project.

Activity	Anirudha	Aditya	Prof Zaheed Shaikh
1. Requirement Gathering			
1.1 Interaction with customer	R	C	A
1.2 Preparing SRS	C	R	A
2. Design			
2.1 Preparing Block diagram	C	R	A
2.2 Writing Functional Requirements	C	C	A

2.3 Writing Non-Functional Requirements	C	C	A
2.4 Developing Use Case	C	C	A
2.5 Developing Test Cases	R	C	A
3. **Planning**	C	C	A
4. **Coding**			
4.1 Dataset & Model Training	C	R	A
4.2 Backend	R	C	A
4.3 Front end/ UI	R	C	A
4.4 Various API to be used	C	C	A
5. **Testing**			

5.1 Accuracy	E	E	A
5.2 Precision	E	E	A
5.3 System Testing	E	E	A

C: Creator, R: Reviewer, A: Approver E: Executor

We even prepared a timeline chart which will help to assign resources, deadlines etc. This will also help in monitoring and tracking the progress of the project.

Date	Task	Description
02/01/2024 - 05/01/2024	Lit. Survey and Topic Selection	Reviewing literature and selecting a research topic for focused study.
08/01/2024 - 12/01/2024	Lit. Survey and Topic Selection	Reviewing literature and selecting a research topic for focused study.
15/01/2024 - 19/01/2024	Problem definition	Defining the core issue or challenge to be addressed in the research.
22/01/2024 - 26/01/2024	Experiment 1	Making an SRS document for the same problem statement.
29/01/2024 - 02/02/2024	Design and Plan	Developing a comprehensive design and plan for the specified project or

		initiative.
05/02/2024 - 09/02/2024	Design and Plan (Experiment 2)	Developing a comprehensive design and plan for the specified project or initiative.
12/02/2024 - 16/02/2024	Prototyping	Building a prototype mobile app for efficient disease detection and real-time crop monitoring in agriculture.
19/02/2024 - 23/02/2024	Prototyping	Building an efficient prototype for GastroIntestinal Disease Detection
26/02/2024 - 01/03/2024	ISE	Break
04/03/2024 - 08/03/2024	Implementation	Executing or putting into action a plan, project, or solution. Work on creating functionality of WebApp. Work on training the model for higher accuracy in prediction.
11/03/2024 - 15/03/2024	Progress evaluation	Assessing the progress made in a project or task to gauge development and identify areas for improvement.
18/03/2024 - 22/03/2024	Implementation	Executing or putting into action a plan, project, or solution. Work on creating functionality of

		WebApp. Work on training the model for higher accuracy in prediction.
25/03/2024 - 29/03/2024	Implementation (Experiment 3)	Executing or putting into action a plan, project, or solution. Work on creating functionality of WebApp. Work on training the model for higher accuracy in prediction.
01/04/2024 - 05/04/2024	Testing	Testing the Integration of components of the WebApp.
08/04/2024 - 12/04/2024	Final evaluation	Concluding assessment or appraisal to determine the overall performance or outcomes.

Gant Chart :

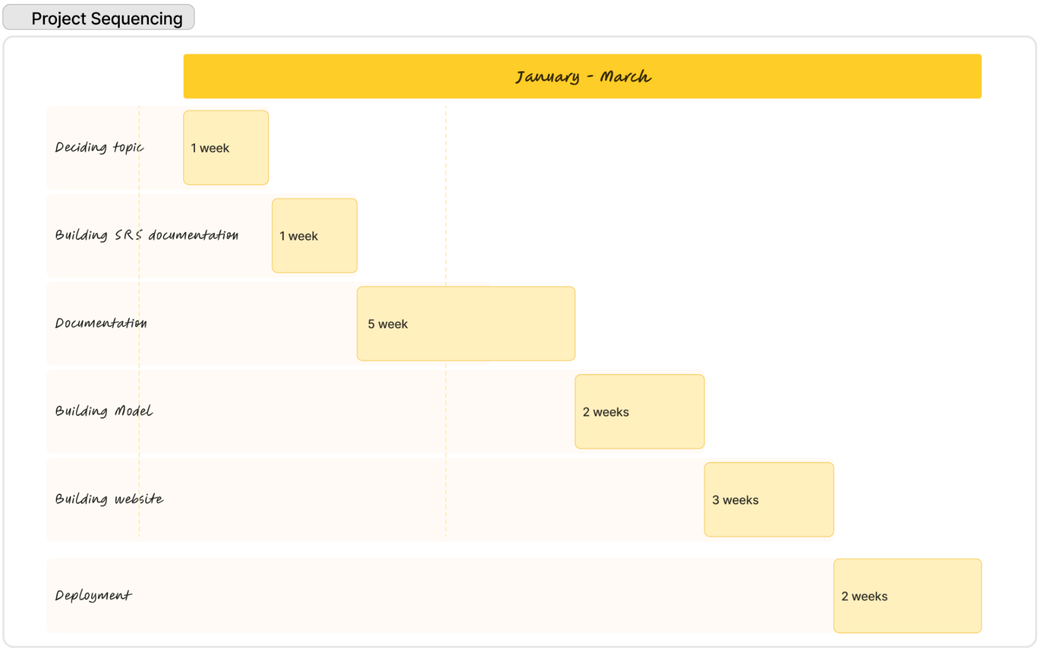


Figure 3

Methodology to be used:

The Agile methodology is chosen for its inherent flexibility and adaptability, allowing teams to navigate the uncertainties and changes common in dynamic project environments. By breaking down the project into small, manageable iterations or sprints, Agile facilitates early and continuous delivery of valuable increments. This iterative approach ensures that a Minimum Viable Product (MVP) is delivered swiftly, enabling stakeholders to realize benefits sooner. The customer-centric nature of Agile methodologies emphasizes continuous collaboration, regular feedback loops, and iterative development, ensuring that the final product aligns closely with user expectations.

We would start by defining high-level Epics such as “Deep Learning Model to perform Gastrointestinal Disease Detection”, “Admin Portal”, “Doctor Portal”, “Image Upload functionality”, “Chat room for doctors”etc. Each Epic can then be broken down into specific User Stories and be implemented in sprints.

These User Stories from the Product Backlog, prioritized based on factors like user value and dependencies. The project would be executed in iterative sprints, typically lasting 2-4 weeks, with each sprint focusing on a subset of prioritized User Stories. Daily stand-up meetings would ensure continuous communication, and regular demos at the end of each sprint would showcase completed features to stakeholders, allowing for feedback and adjustments. The Agile approach enables flexibility, continuous collaboration, and responsiveness to evolving requirements, ensuring the successful delivery of a multifaceted platform that caters to user needs and fosters community engagement.

3.3 Software Design Document (All applicable diagrams)

Figure 4 : Use case diagram

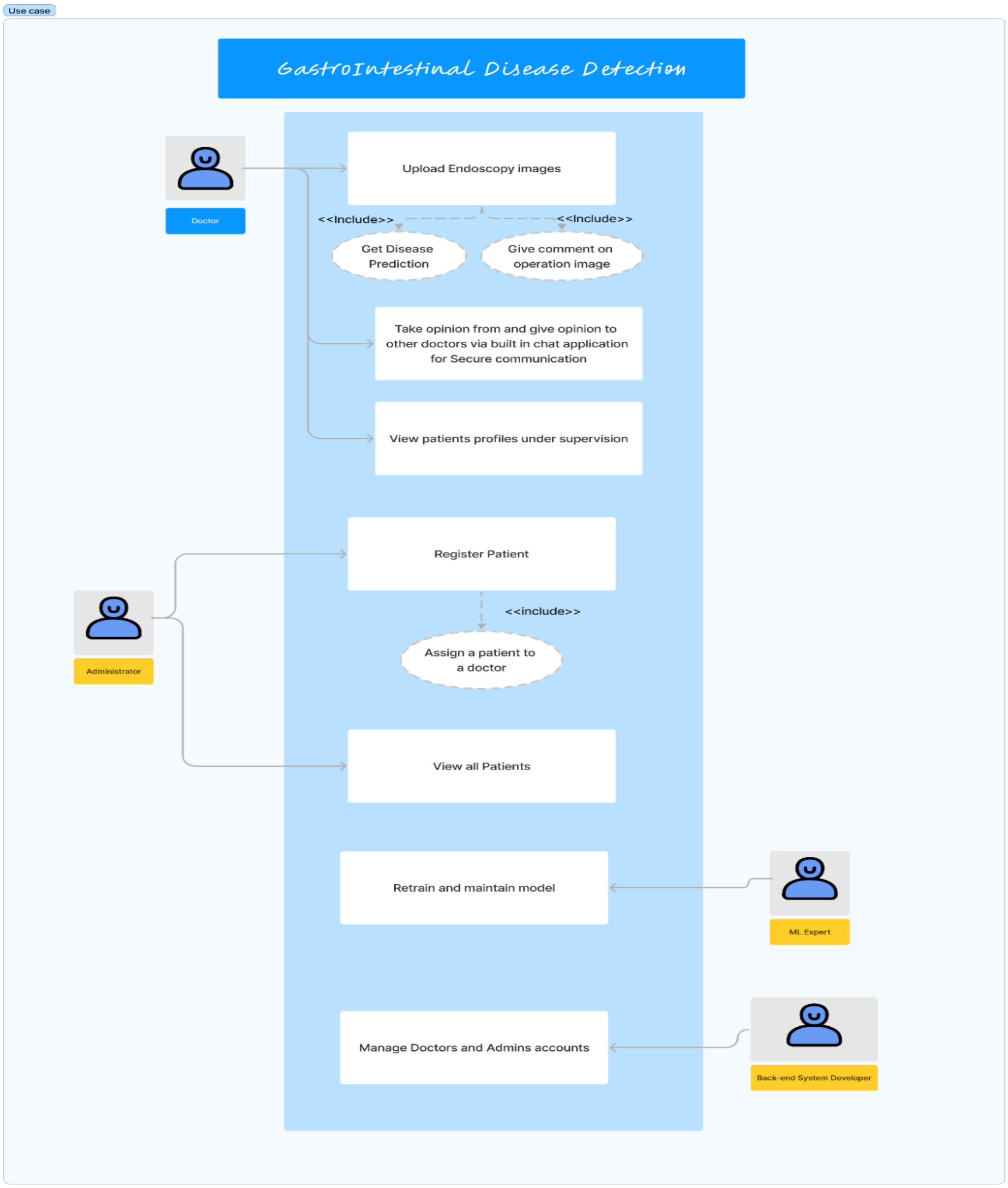

Use case
GastroIntestinal Disease Detection
Doctor
Upload Endoscopy images
<<Include>>
<<Include>>
Get Disease Prediction
Give comment on operation image
Take opinion from and give opinion to other doctors via built in chat application for Secure communication
View patients profiles under supervision
Administrator
Register Patient
<<include>>
Assign a patient to a doctor
View all Patients
Retrain and maintain model
ML Expert
Manage Doctors and Admins accounts
Back-end System Developer

Figure 4

Figure 5 : Class Diagram:

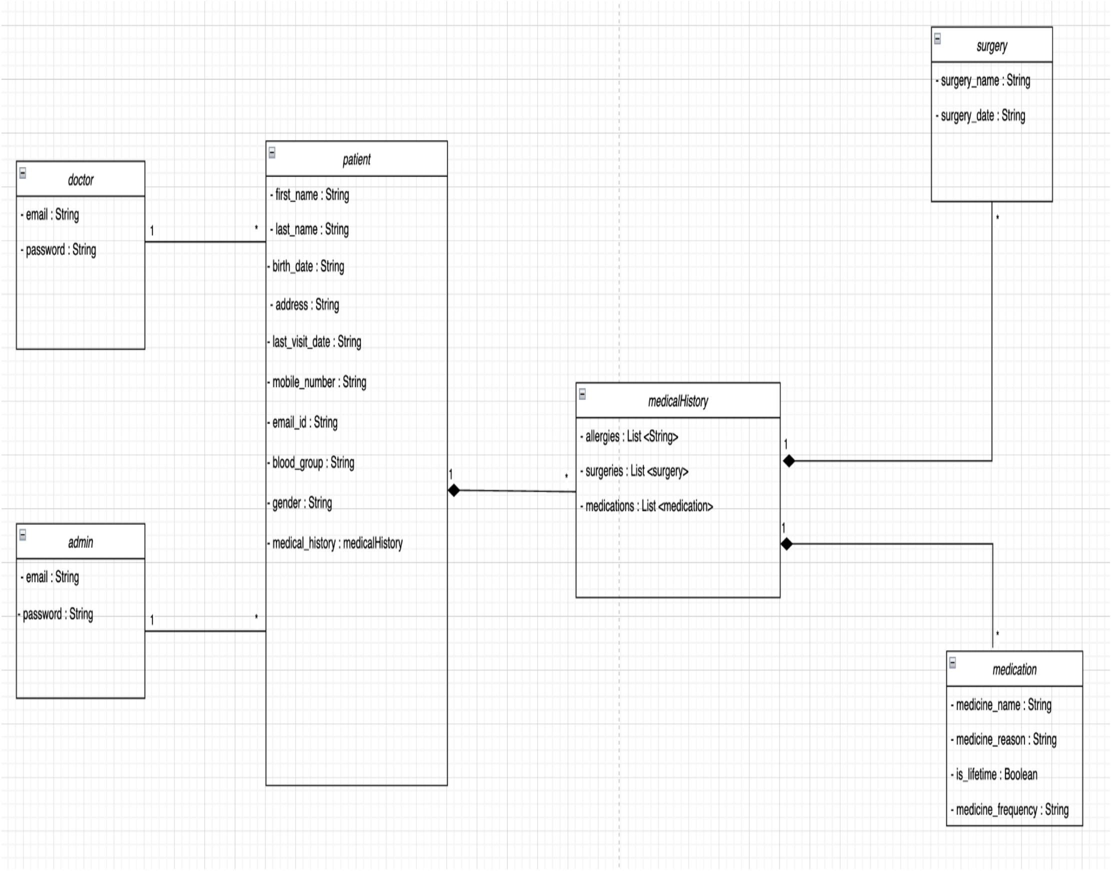


Figure 5

Figure 6 : Sequence Diagram:

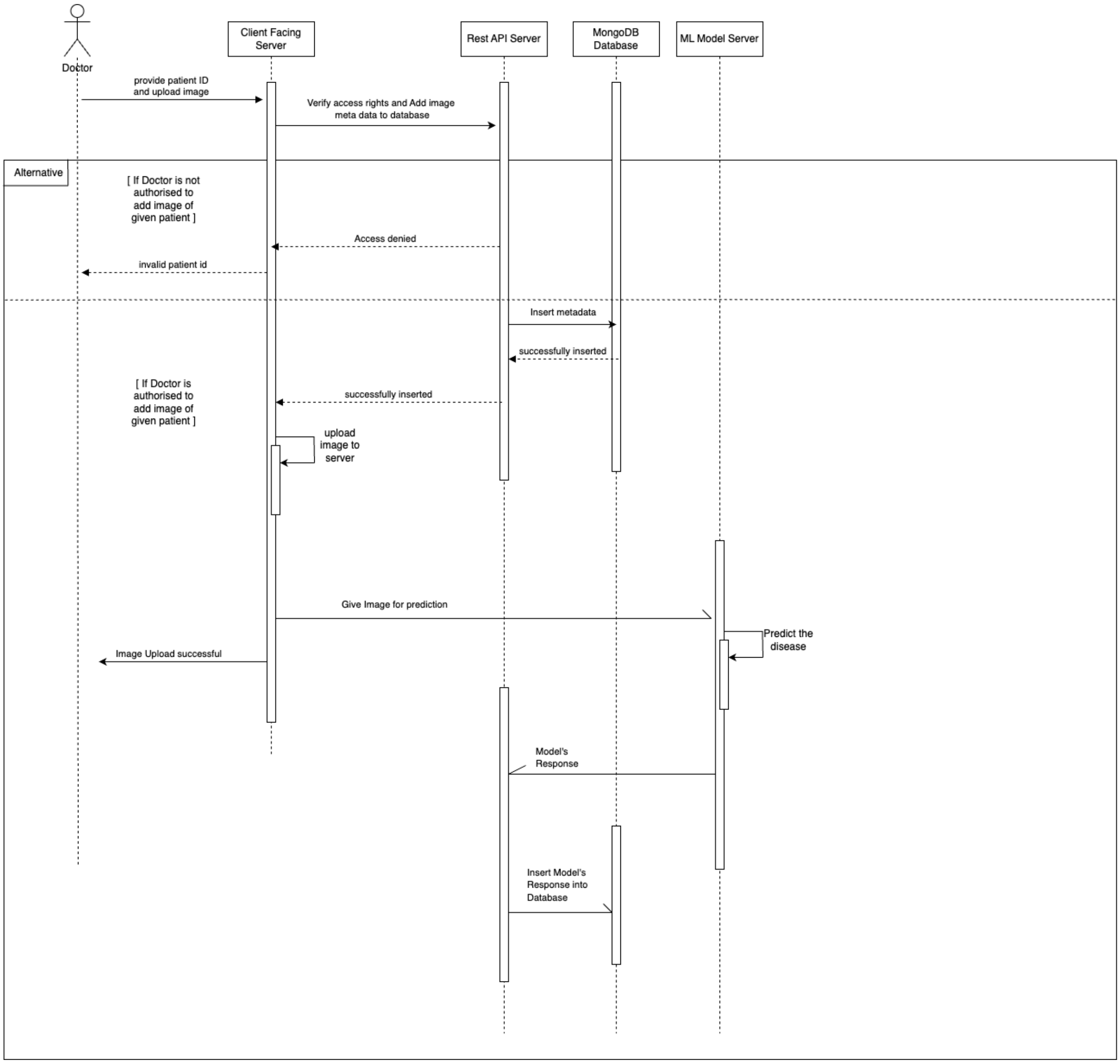
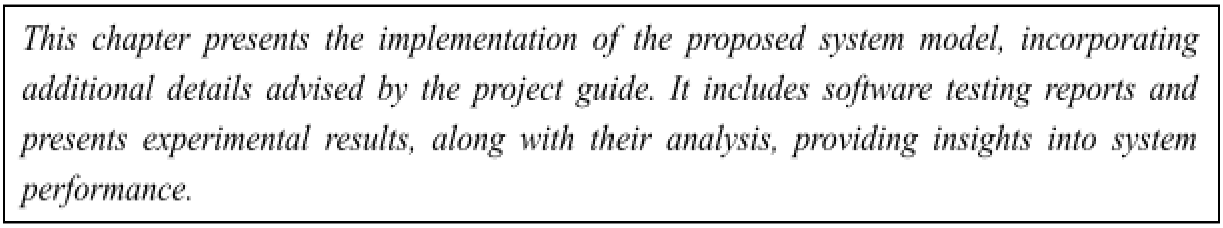

Doctor
Client Facing Server
Rest API Server
MongoDB Database
ML Model Server
provide patient ID and upload image
Verify access rights and Add image meta data to database
Alternative
[If Doctor is not authorised to add image of given patient]
Access denied
invalid patient id
Insert metadata
successfully inserted
[If Doctor is authorised to add image of given patient]
successfully inserted
upload image to server
Give Image for prediction
Predict the disease
Image Upload successful
Model's Response
Insert Model's Response into Database

Figure 6

Chapter 4

Implementation and experimentation of one of the issues related to project work topic



This chapter presents the implementation of the proposed system model, incorporating additional details advised by the project guide. It includes software testing reports and presents experimental results, along with their analysis, providing insights into system performance.

4.1 Proposed system model implementation

ML Model Architecture Diagram :

- VGG (Visual Geometry Group) is a model that has a simple yet deep and efficient architecture. The model consists of :

    1. MaxPooling layers of filter size 2x2 with a stride 2 resulting in halving of the size of the image
    2. Convolution layer of filter size 3x3 with padding same and stride one.
    3. While Transitioning from one group of Convolutional layer to another a MaxPooling layer is used that reduces the image size for the Convolutional layer to operate on

- The Activation function used here is ReLU activation function as ReLU helps in alleviating the vanishing gradients problem to some extent.
- The top layers consists of dense layers with drop out to prevent overfitting and the final layer has soft max activation function for the final classification
- The loss function used is Categorical CrossEntropy with Optimizer as Stochastic Gradient Descent with learning initial learning rate as 0.001
- The VGG19 has its weights pre initialized to the imagenet weights to avoid random initialization and decrease the training time
- For testing the model the dataset is divided into 80:20 (Training to Testing

ratio). The metrics used to evaluate the model are F1 Score, Accuracy, Precision

Why Choose VGG19?

VGG19, a variant of the VGG (Visual Geometry Group) model, has several advantages:

1. **Uniform architecture**: The architecture of VGG19 follows a uniform pattern of stacked convolutional layers with small receptive fields (3x3 filters) and max-pooling layers, which simplifies design and implementation.
2. **Good generalization**: VGG19 performs well on a variety of image recognition tasks. Its deep architecture allows it to capture complex patterns and features in images, leading to good generalization performance.
3. **Feature representation**: The convolutional layers in VGG19 extract hierarchical features from images, which can be utilized for tasks such as image classification, object detection, and semantic segmentation.
4. **Scalability**: VGG19 can be scaled up or down depending on the computational resources available and the specific requirements of the task. This scalability makes it adaptable to different hardware configurations and application scenarios.

5. **Combat Vanishing gradient:** The use of ReLU helps in combating the vanishing gradient problem by making use of ReLU activation function

4.2 Inclusion of Any additional details as suggested by Project Guide/during progress seminar

We appreciate the suggestion provided, the additional details and feedback received during the progress seminar and from the project guide.

1. One major feedback that was provided was to increase the accuracy of the deep learning model used. The earlier model used had an accuracy of 14% but now by exploring other deep learning model, VGG19 was chosen and used resulting in a competent accuracy of 78%

2. Earlier the model was not integrated into the system but now it’s asynchronously integrated with the website. The doctor can upload images to the server and if the doctor has the access to do so, the image will be added to the server and the ML Model Server will simultaneously compute the response. In some time the result will be visible to the doctor by viewing the patient’s profile.

4.3 Software Testing (Software testing reports at various levels)

1. Developer testing :

    1. Unit Testing: In the Unit Testing Individual components of the software are tested eg: Whether files are being uploaded properly, The Deep Learning model is able to handle images when many users are using the model simultaneously. So basically in unit testing each microservice of the WebApp is tested individually to ensure they run correctly.
    2. Integration Testing: In Integration testing we make sure that all the individual parts are working as expected when put together in a system. eg.: File uploading works when integrated with a chat room so that doctors can chat and share images. Thus integration testing checks whether the different microservices are able to correctly integrate and work together.

3. Code Reviews: The code is continuously reviewed and tested to make sure that all the code remains readable and is working properly at all times

2. Alpha/Beta Testing:

1. Fostering Regular feedbacks from doctors in order to make sure that the accuracy of the model is good and is performing well under different circumstances
2. Taking Feedback from the admin to ensure that the website is helping in comfortable administration of the hospital and that the chats of the doctors can be monitored comfortably
3. Taking feedback from doctors too ensure ease of collaboration with other doctors

3. User Acceptance Testing (UAT):

a. Ensure that the website has high usability by making sure that the website can be accessed through screen readers

b. Evaluating the accuracy of the machine learning model's predictions using a separate test dataset

c. Make sure that the model isn't overfitting

These scenarios and report details are more specific to the project's domain and components, including the machine learning model and the website. They cover a range of testing aspects, from functional and non-functional testing to user feedback and real-world usage scenarios.

4.4 Experimental results and its analysis

After training the model , the total accuracy and precision achieved by the model is 78%. The F1 Score obtained by the model is shown in the diagram below. The model performs reasonably well across all the classes.

Figure 7: Figures for metrics of the model:

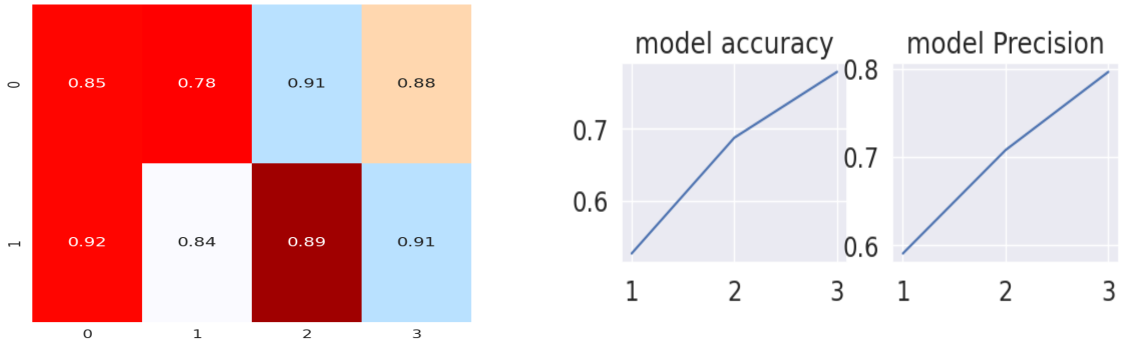


Figure 7

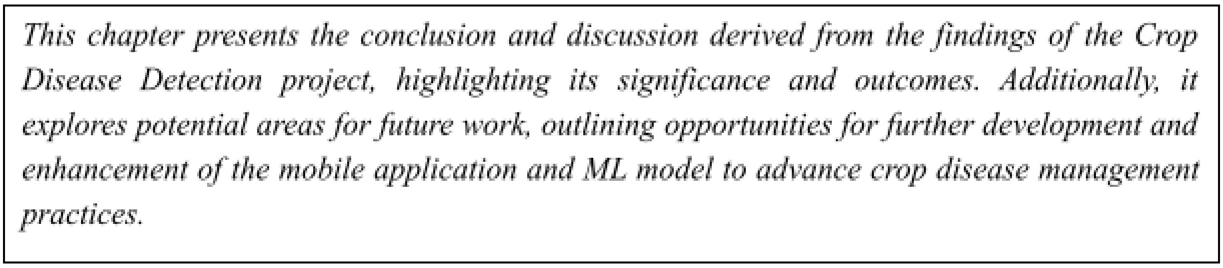
While the model is performing well, it is suspected that overfitting is occurring to some extent. The model architecture needs further improvement in order to combat the overfitting.

Various other models have been tried and experimented. Google's Inception V4 has been tried along with the Inception ResNet Model. Also a combination of Inception, ResNet and SVM has been tried. Among all of these models the VGG19 performs the best due to its simple yet deep feature extracting architecture.

The website has been developed keeping the ease of use of doctors and administrators in mind. The website can handle multiple users using the website simultaneously. MongoDB has been used as the database supporting faster read write times. The Website helps the doctors to get the prediction of the VGG19 model as well as collaborate with other doctors. The admin can monitor chats and register patients comfortably through the website

Chapter 5

Conclusion and Further Work



This chapter presents the conclusion and discussion derived from the findings of the Crop Disease Detection project, highlighting its significance and outcomes. Additionally, it explores potential areas for future work, outlining opportunities for further development and enhancement of the mobile application and ML model to advance crop disease management practices.

Chapter subtopics should be font 14 Times New Roman and 1.5 spacing bold, left aligned and justified

5.1 Conclusions

Thus the proposed integrated system consisting of a website and deep learning model successfully helps the doctors to take second opinion from the DL model as well as collaborate with other doctors within the same hospital. The proposed system also enables the admin to efficiently manage patients and monitor the chats among doctors to ensure no deatils related to the patients are realeased

5.2 Further Work

While the proposed system gives high accuracy, there is overfitting affecting the model's overall performance. Improvements like adding Dropouts strategically between layers and using image augmentation techniques like flipping, shearing rotation etc along with collaborating with multiple hospitals to obtain a larger dataset. Advanced Deep Image processing techniques like Adaptive Median Filtering can be used to process the images and remove reflections caused by endoscopic tubes and thus improve model’s performance. In future a feedback loop can be added to enable the retraining of the model on images that it has misclassified

The website can use cloud managed services like AWS Cognito for user identity creation and management. In future the website can add a feature for the doctors to share their research with other doctors within the same hospital. The website has not been hosted yet but in future hosting services can be used. Cloud based computing instances like AWS EC2 can be used. The images are currently being stored in the server itself but in future cloud based solutions like AWS S3 Object Storage can be used to store images on the cloud.

Thus the proposed solution is more on-premise focused (since the focus was on implementing microservice architecture with smooth integration of each microservice into the WebApp) which can later migrate to cloud.

Bibliography

- Segu, S., Drozdzal, M., Pascual, G., Radeva, P., Malagelada, C., Azpiroz, F., & Vitrià, J. (2016). Generic feature learning for wireless capsule endoscopy analysis. *Computers in Biology and Medicine*, 79(1), 163–172. https://doi.org/10.1016/j.compbiomed.2016.10
 PubMed Web of Science ®Google Scholar

- Gamage, C., Wijesinghe, I., Chitraranjan, C., & Perera, I. (2019). GI-Net: Anomalies classification in gastrointestinal tract through endoscopic imagery with deep learning," MERCon 2019 - Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference, 66–71, https://doi.org/10.1109/MERCon.2019.8818929
 Google Scholar

- Khan, M. A., Majid, A., Hussain, N., Alhaisoni, M., Zhang, Y., Kadry, S., & Nam, Y. (2021). Multiclass stomach diseases classification using deep learning features optimization. *Computers, Materials & Continua*, 67(3), 3382–3398. https://doi.org/10.32604/cmc.2021.014983
 Web of Science ®Google Scholar

- VGG 19 model reference : https://arxiv.org/pdf/1409.1556.pdf

- Owais, M., Arsalan, M., Choi, J., Mahmood, T., & Park, K. R. (2019). Artificial intelligence-based classification of multiple gastrointestinal diseases using endoscopy videos for clinical diagnosis. *Journal of Clinical Medicine*, 8(7), 986. https://doi.org/10.3390/jcm8070986

  PubMed Web of Science ®Google Scholar

Acknowledgement

We express our sincere gratitude to all those who have contributed to the successful completion of this project. Their guidance, support, and encouragement have been invaluable throughout this endeavor.

First and foremost, we would like to extend our deepest appreciation to our esteemed project guide, Prof. Zaheed Shaikh, for his unwavering guidance, expert advice, and continuous support. His wealth of knowledge and expertise have been instrumental in shaping this project and helping us navigate through the challenges we encountered. We are truly grateful for his mentorship and the invaluable lessons we have learned under his tutelage.

We are grateful to the open-source community for providing access to valuable resources, datasets, and tools that have facilitated our research and development efforts. The availability of these resources has been instrumental in advancing our understanding and enabling us to contribute to the field.

Finally, we would like to express our gratitude to our families and friends for their unwavering support, patience, and motivation throughout this journey. Their belief in our abilities and their constant encouragement have been a driving force, enabling us to persevere and overcome the challenges we faced.

We sincerely appreciate the contributions of all those who have directly or indirectly supported us in this endeavor. Their involvement has been invaluable, and we are truly grateful for their assistance in making this project a success.