Gastrointestinal Disease Detection

Anirudha Ta : 16010121199

Aditya Ved : 16010121208

Why Choose GI Disease detection ?

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 tumours 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.

4) An automated report generation based on the ebdoscopic results will help the doctors give more time to care for the patients

Improvements Suggested

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.

Tech Stack Used :

- Backend :
  1. Database : MongoDB
  2. Fast Api

- Front End : HTML, CSS, JavaScript, JQuery

- Machine Learning Model :
  1. Tensorflow/Keras
  2. Dataset : Kvasir

Literature Review

Paper	Dataset Used	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%

Literature Review

Khan et al.,	Kvasir	The ulcer annotated images are utilized to train the Mask RCNN model to obtain output in the form of 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.	99.13 %
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%

Dataset Used

1 The dataset used for the ML model is Kvasir Dataset developed by Vestre Viken Health Trust (VV) in Norway.

2) The images in the dataset are divided into 6 classes with 500 images in each category

- Esophagitis
- Polyps
- Ulcerative-colitis
- Normal-z-line
- Normal pylorus
- Normal cecum

3) The resolution in the images ranges from 720x576 up to 1920x1072 pixels. In some images there is shown arrangement of endoscope inside intestine

4) Each of the classes have a CSV file containing useful information like AutoColorCorrelogram, PHOG, Edge Histogram. These metadata help in more accurate detection of tumour

Dataset Used

1 The dataset used for the ML model is Kvasir Dataset developed by Vestre Viken Health Trust (VV) in Norway.

2) The images in the dataset are divided into 6 classes with 500 images in each category

- Esophagitis
- Polyps
- Ulcerative-colitis
- Normal-z-line
- Normal pylorus)
- Normal cecum

3) The resolution in the images ranges from 720x576 up to 1920x1072 pixels. In some images there is shown arrangement of endoscope inside intestine

4) Each of the classes have a CSV file containing useful information like AutoColorCorrelogram, PHOG, Edge Histogram. These metadata help in more accurate detection of tumour

Classification Model Used :

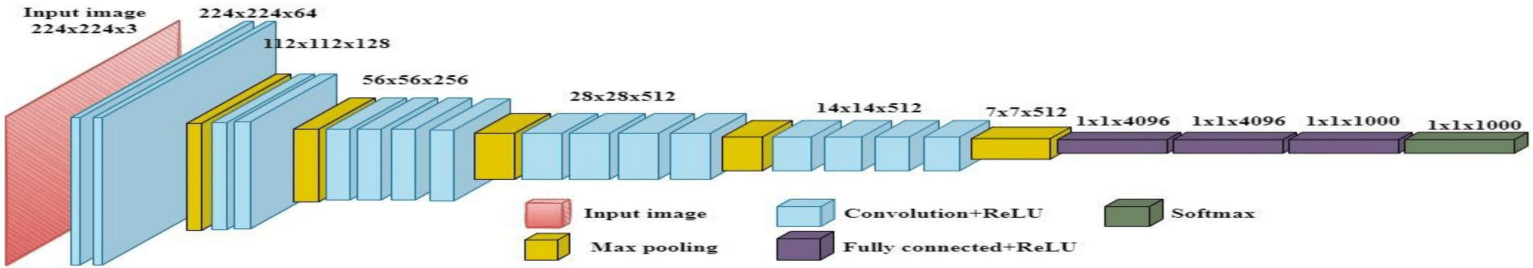
ML Model Architecture Diagram :

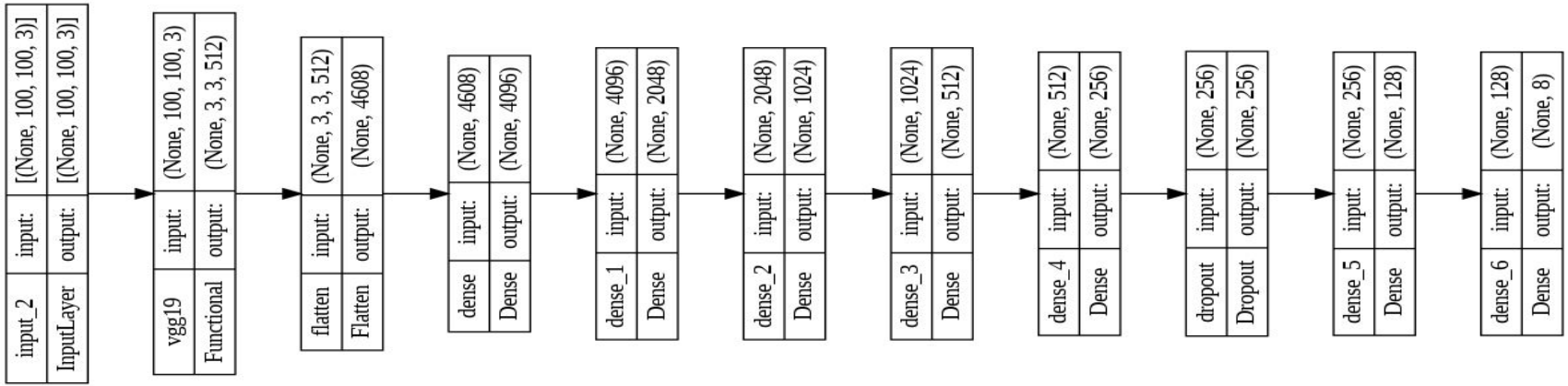
- VGG (Visual Geometry Group) is a model that has a simple yet deep and efficient architecture. The model consists of :

    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

Various Parts of the Model:





Key Aspects of WebApp

1. Microservices Architecture with the below microservices:
    a. Rest API Server
    b. Client Facing Server
    c. Authentication System Server
    d. Machine Learning Model Server
2. Client Side Rendering Architecture
3. Token Based Authentication System using Asymmetric Key encryption to sign and verify JWTs (JSON Web Tokens)
    a. Setting Authorization Headers
    b. Setting cookies
4. Asynchronous integration of machine learning model with WebApp.
5. Architecture Diagram of how the microservices of the system interacts with each other.

Brief explanation of each Microservice of WebApp

1. Rest API Server:

This server serves all the data (by fetching from DB) using Rest-API's.. It does the authorization of data access by reading the access token from the authorisation header of the request. The token also contains the user type in the payload. This is how the authentication of different users is performed. Each route "Depends" on auth module created in the server to verify the authorisation header [Concept of Dependency Injection utilised here]

This server has 4 main layers.

a. *The 1st layer is of the Pydantic Models. They ensure that the correct format of data is present in the HTTP Body.*
b. *The 2nd layer is of authorisation.*
c. *The 3rd layer is the application logic within the route.*
d. *The 4th layer is Data Abstraction which connects to the database and gets the data (by ensuring safely opening and closing the connection)*

2. Client Facing Server:
This server serves all the web pages and static content demanded by the end user. This server is the point of contact of the Web App with the end user (admins and doctors). This server also stores the images of the patients. This server also makes some calls to the Rest-API Server to fetch some data to perform some app logic or render some dynamically generated templates using jinja. This server is responsible for logging out the user by inserting the token into logged out tokens database. This server is responsible for sending the image link to the ML/DL Model Server which does the prediction and then sends the response to Rest-API Server to insert into the Database.

Brief explanation of each Microservice of WebApp

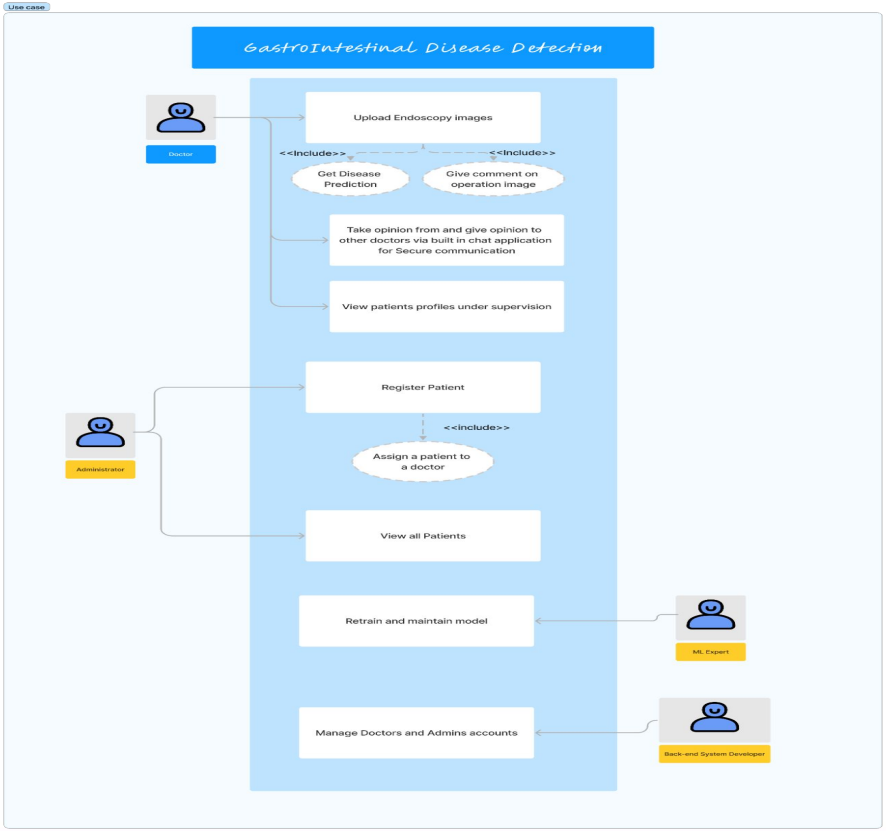
3. Authentication System Server:

This auth server issues tokens after verifying credentials. This token is stored in the client's browser and used every time the browser makes a request to the Client facing server or Rest-API Server. Only this microservice server in the entire application possess the private key to sign the token. Other modules of the system verify the token by checking if the token is signed using the private key of authentication microservice. This verification is done using Public key. Authentication of different types of users is performed by viewing the "user_type" in access token which is present in the payload of the token. Only This microservice has access to the user credentials Database which stores the email and Password. So this microservice receives login credentials and then verify them and then issue a token if the credentials are correct. Once the token is logged out or expired : the other modules of the application will redirect to login page and this is how the authentication microservice is integral part of the application.

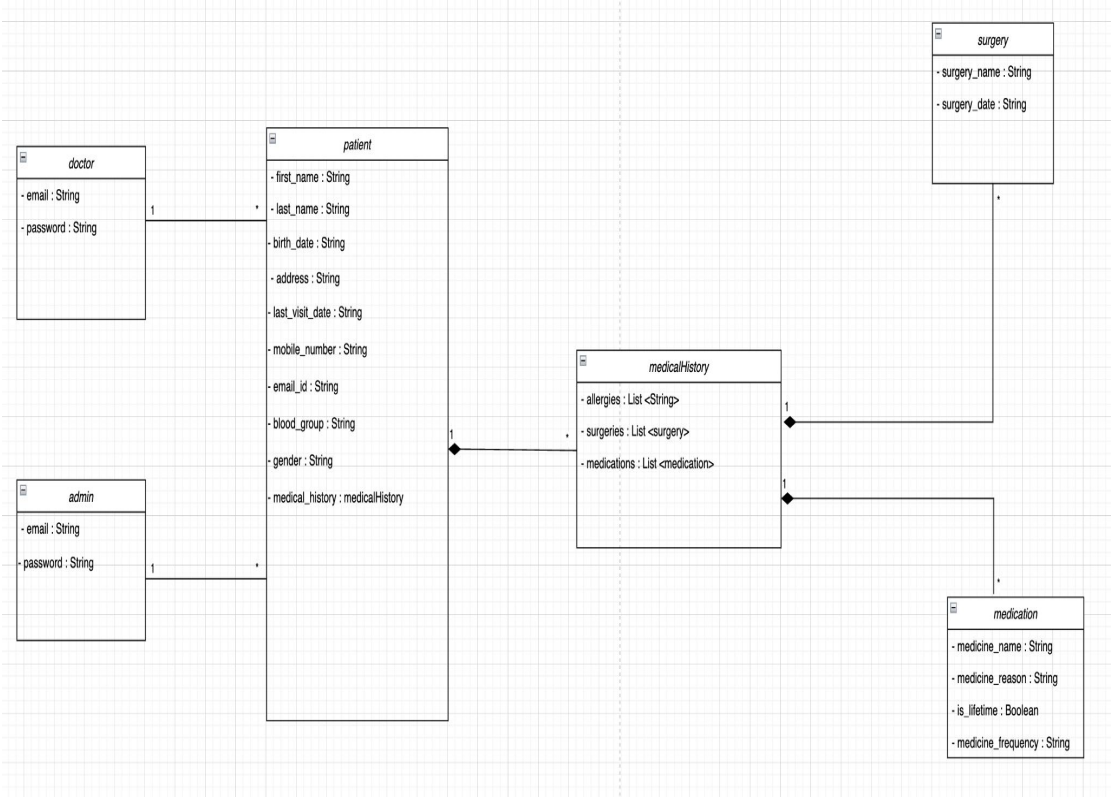
4. Machine Learning Model Server:

This server runs the Deep Learning model to do prediction. The purpose of this server is to run gastrointestinal disease detection. This server accepts the link of the image and then calls the client facing server to fetch the image and then perform prediction on it. After the prediction is complete this server will call the rest api server to save the model's response in the database. The model is loaded in this server's memory when the server starts so that each request can be served fast by just calling the model and doing the prediction.

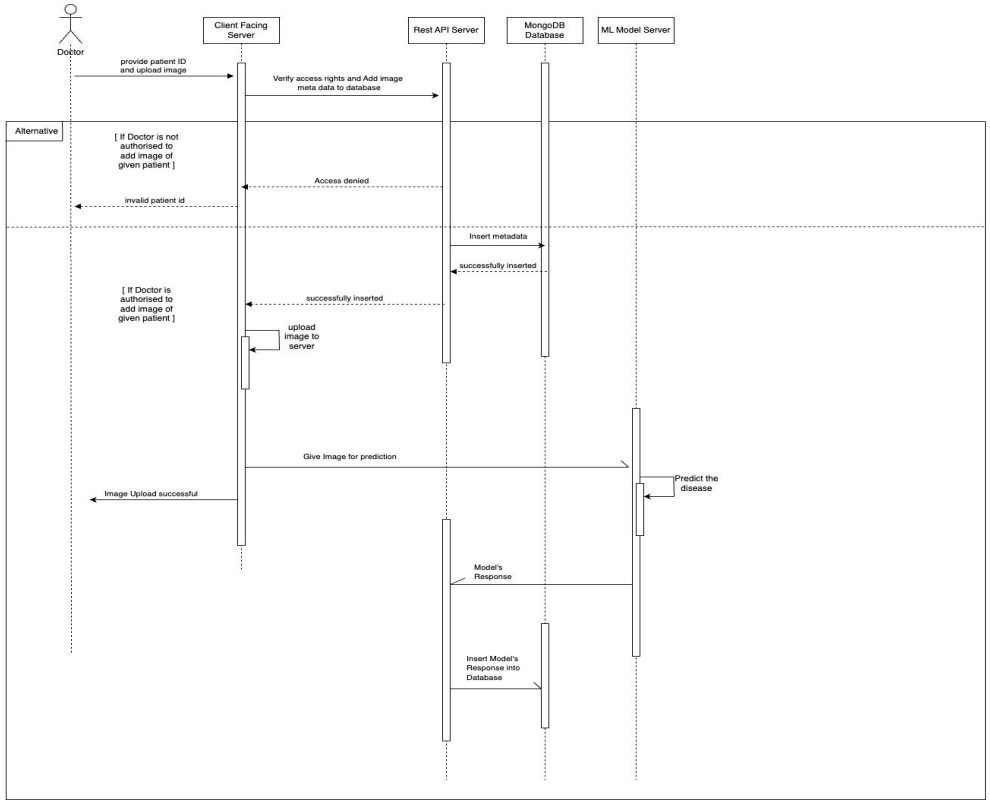
Use Case Diagram



Class Diagram



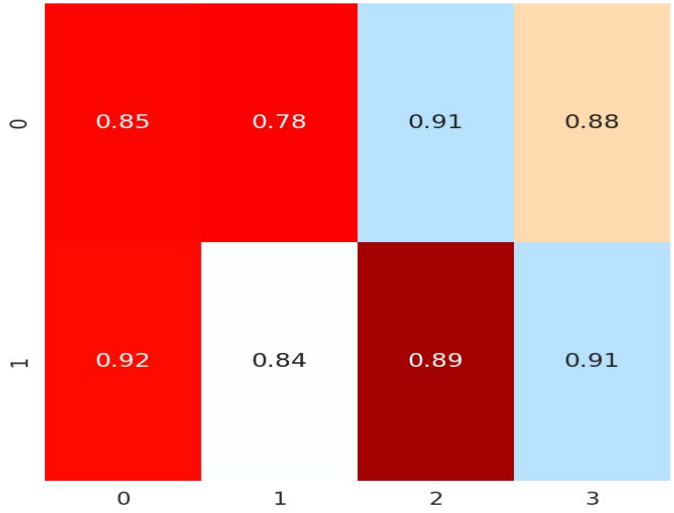
Sequence Diagram

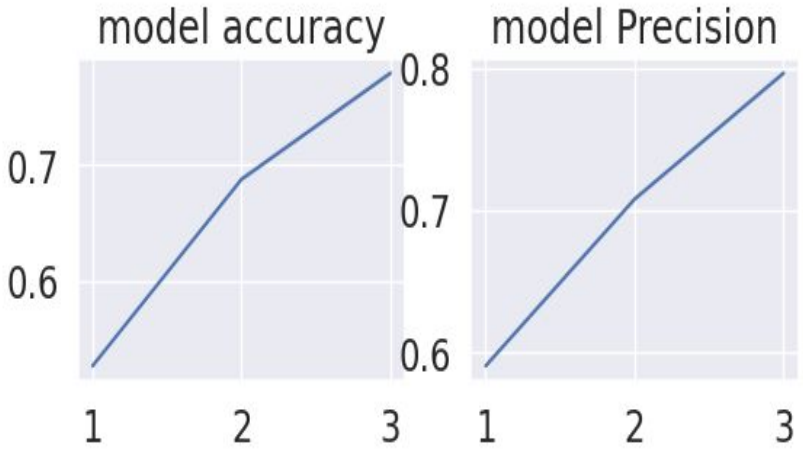


Results :

- 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.
- 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

Results


0.85
0.78
0.91
0.88
0.92
0.84
0.89
0.91
0
1
0
1
2
3


model accuracy
0.7
0.6
1
2
3
model Precision
0.8
0.7
0.6
1
2
3

Conclusion and Future Scope

- 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 details related to the patients are released
- 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.
- 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.

Thank You !!