DIAGNOHUB

Overview

This repository contains a suite of Python scripts that together form an AI-powered healthcare assistance system. This system integrates a chatbot for symptom querying, geolocation services for emergency aid, and a lung cancer prediction model using TensorFlow. The application is wrapped in a user-friendly GUI created with `tkinter`.

Components

- `Chatbot_Logic.py`: Powers a chatbot capable of responding to health-related queries.
- `Geolocation_OOP.py`: Provides geolocation functionalities to locate nearby healthcare services.
- `tensorflow_lung_cancer_model.py`: Contains a TensorFlow model for predicting lung cancer from medical images.
- `GUI.py`: A graphical user interface that integrates all the above components.

Installation

Prerequisites

- Python 3.x
- Pip (Python package manager)

Dependencies

Install the necessary Python packages by running:

```
pip install -r requirements.txt
```

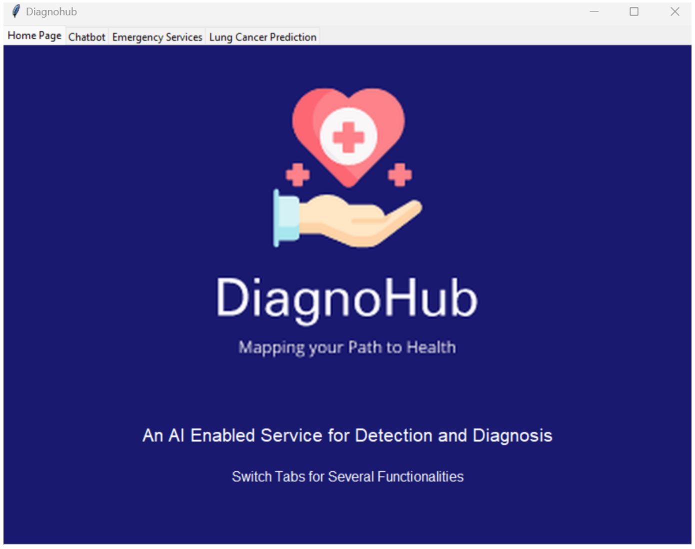
Usage

Running the Application

Navigate to the repository directory and run:

```
python GUI.py
```

This will open the main application window where you can interact with the different functionalities offered by the system.

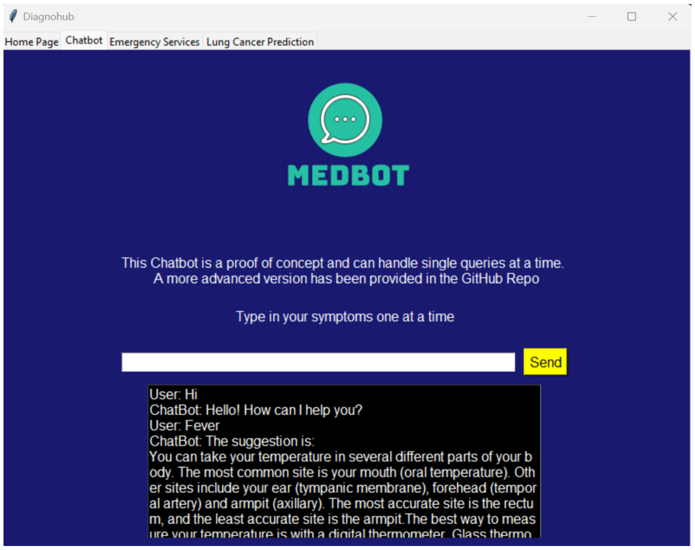


DiagnoHub

Mapping your Path to Health

An AI Enabled Service for Detection and Diagnosis

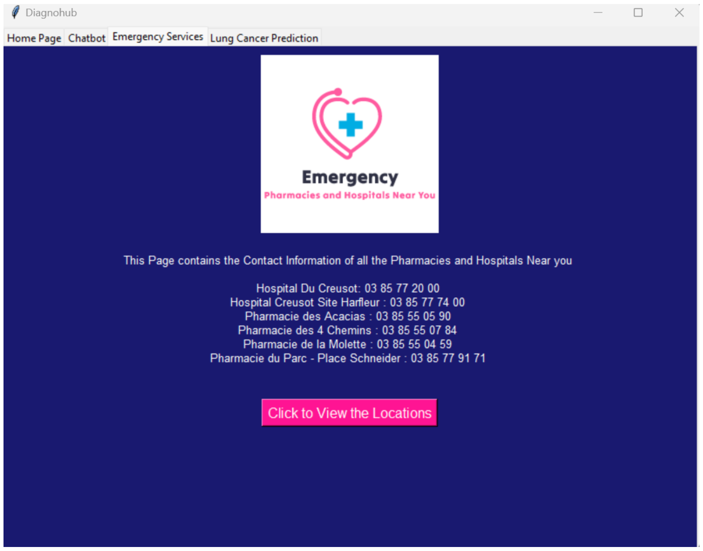
Switch Tabs for Several Functionalities



This Chatbot is a proof of concept and can handle single queries at a time.
A more advanced version has been provided in the GitHub Repo

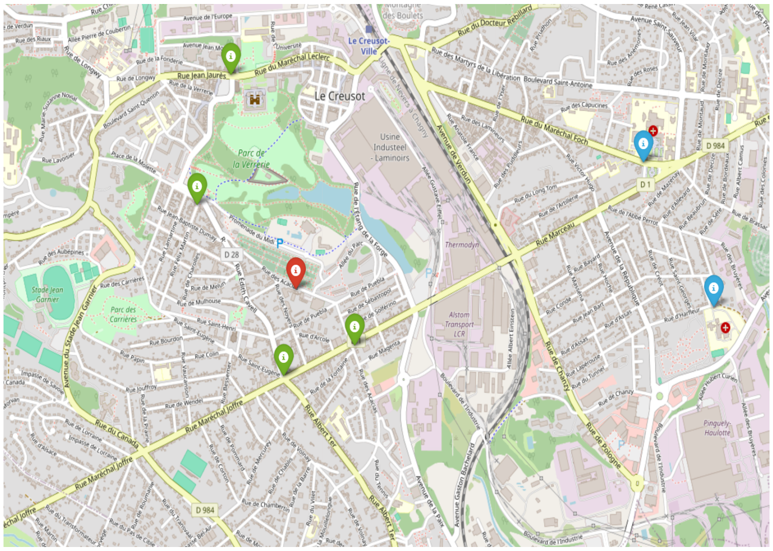
Type in your symptoms one at a time

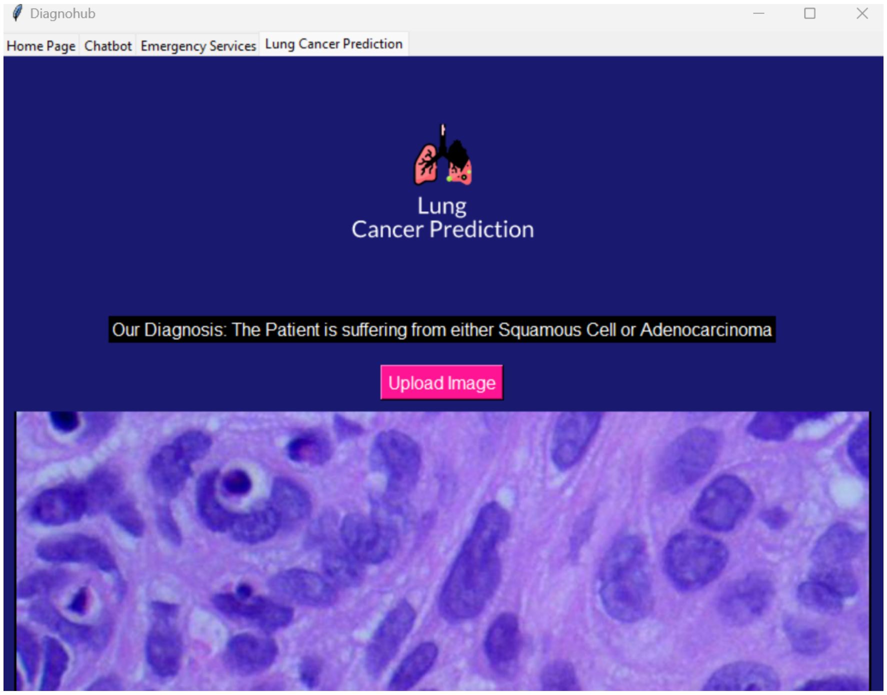
```
User: Hi
ChatBot: Hello! How can I help you?
User: Fever
ChatBot: The suggestion is:
You can take your temperature in several different parts of your b
ody. The most common site is your mouth (oral temperature). Oth
er sites include your ear (tympanic membrane), forehead (tempor
al artery) and armpit (axillary). The most accurate site is the rectu
m, and the least accurate site is the armpit.The best way to meas
ure your temperature is with a digital thermometer. Glass thermo
```



This Page contains the Contact Information of all the Pharmacies and Hospitals Near you

Hospital Du Creusot: 03 85 77 20 00
Hospital Creusot Site Harfleur : 03 85 77 74 00
Pharmacie des Acacias : 03 85 55 05 90
Pharmacie des 4 Chemins : 03 85 55 07 84
Pharmacie de la Molette : 03 85 55 04 59
Pharmacie du Parc - Place Schneider : 03 85 77 91 71





Our Diagnosis: The Patient is suffering from either Squamous Cell or Adenocarcinoma

Chatbot

The Medbot is a chatbot custom-made for answering any medical-related queries based on user input. The code consists of a web crawler that can perform term-based searches using the request library. The Request Library fetches information from Google using the Google Search API and returns the list of URLs from them. Since MedBot is used for medicine-related queries, the chatbot logic is written in such a way, that the list of URLs returned refers to sites containing information on Medical Symptoms. We scrape data from the "Management and Treatment" section of my.clevelandclinic.org using the beautiful soup library. The Chatbot provides information on how to treat any symptoms that the user provides as input. The advanced version of the chatbot uploaded as Search_API.ipynb uses Named Entity Recognition by a Spacy model en_ner_bc5cdr_md, trained on medical terms like fever, headache, myocardial infarction, etc. The model detects the symptoms from a sentence provided by the user, for example, "I am suffering from Fever, Headache, and vomiting", the model will detect fever, headache, and vomiting make a list of strings, and feed them one by one to the Chatbot. The Chatbot on receiving these terms will do a query-based search using Google Search API and fetch the results using the requests Libary. The Rest of the procedure works according to what is explained.

Geolocation Services

Access the geolocation tab to view nearby healthcare facilities.

- The application uses predefined locations, which can be modified as per your requirements, the purpose is to create an interactive map displaying the user's location and nearby healthcare facilities. The application allows users to customize and add predefined locations, represented by markers of different colors, to the map. It utilizes the Geopy library for geolocation services and Folium for map visualization.

How It Works:

The program initializes a LocationMap class, which encapsulates geolocation-related functionalities.The user's geolocation is obtained using the Geopy library based on the provided coordinates (latitude and longitude). Additional locations, such as healthcare facilities, can be added using the add_location method. Each location is stored with a name, latitude, longitude, and an optional marker color (default is blue). The program creates a Folium map centered at the user's location using the create_map method. Markers are added to the map for both the user's location (in red) and other predefined locations. Marker colors can be customized, and additional locations like pharmacies or medical posts are represented by green markers. The generated map is saved as an HTML file and displayed directly within the notebook using IPython's display function.

The program is used to visualize the user's location and nearby healthcare facilities, specifically pharmacies. The map is customized with different marker colors for the user's location (red) and pharmacies (green). The program requires the installation of external libraries: Geopy, Folium, and Cartopy. Additionally, the Basemap library is installed via Conda.

Lung Cancer Prediction

The project incorporates the use of convolutional neural networks to train a lung cancer dataset Link:
https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images
Consisting of 5000 images of three classes of Lung Cancer. The classes are

1. Normal Lung with no cancer
2. Adenocarcinoma
3. Squamous Cell Carcinoma.

A custom-made convolutional Neural Network has been made and trained with the images provided to perform a binary class classification after loading the model with tensorflow between, normal lung and cancerous lung.
Initially, libraries and datasets are imported to provide the necessary tools and data for the task. Following this, data visualization techniques are applied to gain insights into the dataset. A subset of the data is reserved for validation, which is crucial for later assessing the model's performance.

In parallel, the training data undergoes image data processing, which includes steps such as normalization and augmentation to enhance the model's learning. The development of the model involves constructing a simple CNN comprising three convolutional layers with max pooling to extract features from images, and two fully connected layers to interpret these features. The architecture is completed with an output layer that employs soft probability to output the predictions.

The model's performance is evaluated through various methods. Loss and accuracy are visualized epoch by epoch to track and refine the training process. Furthermore, the model's predictive accuracy is detailed through the creation of a confusion matrix and a classification report for the validation data. This comprehensive approach ensures a robust build and assessment of the CNN for image classification tasks, starting from preparatory steps to the final evaluation of the model's predictive capabilities.

- In the lung cancer prediction tab, you can upload medical images.
- The TensorFlow model will analyze the image and provide a diagnostic output.

Contact

- Souparno Chattopadhyay: souparno.chattopadhyay@gmail.com
- Sumeet Dash: sumeet.dash772@gmail.com
- Jose Sebastian Ibarra Arregui : sebastian.ibarra@pucp.edu.pe
- Tuhinangshu Gangopadhyay: tuhinangshu.ganguly@gmail.com
- Madinabonu Akramova: madinaakramova@gmail.com
- Daniel Alejandro Reyes Barrios: daniel.reyesbs@hotmail.com