Team Inferno

Topic : HealthTech

Documentation

Overview:

A website that enables a patient to upload his/her details and health reports in a centralized system will then analyze using an advanced Machine Learning model in the backend and the result will be reflected on the website. In case there is a prediction of the presence of disease in the patient, then he/she can directly consult the doctor from our website.The doctor can see the prediction of the model as well as the health report of the patient and the doctor can prescribe medications based on reports. There is no need for the patient to reach the doctor to get medications and they can directly print the prescription from our website.

Solution to Major Problems:-

1) Rush in hospital is a major issue faced in India. After implementing this idea on a large scale, this situation can be controlled. And after the outbreak of Covid-19 across the globe, it will help not only the Patients but also the life saviour Doctors.
2) This idea can also be implemented for research purposes.
3) People from weak financial backgrounds will get an additional benefit with the use of this site as detecting a disease is free of cost as this model is implemented using an advanced Machine Learning Model.

Working:

At the homepage of the website, after logging into the patient's end, the patient can upload the MRI scan and additional health reports which will then be fed into our machine learning model by the backend, which will predict the chances of the presence of the brain tumor in the patient. If the patient is healthy, the model will tell so and there is no need to consult the doctor. If there is any chance of the disease, it will alert the patient and recommend to consult the doctor from our website itself.

Doctor will receive an alert if there is any request from the patient and the doctor can login from the website. After viewing the scanned report and previous health records, the doctor will give prescription and medication based on the conditions, which can be downloaded straight away by the patient from the website.

Now coming to the Machine learning model, it has been trained on brain tumor MRI images which consists of 2225 images for training and 253 images for testing belonging to two classes. Our model consisted of 5 CL and 7 AL layers with 1,131,250 trainable parameters and they achieved highest validation accuracy of 97.23% and also had validation loss of 6.88% on binary classification of the dataset. Performance of the model was stable and good through all the run-time where every epoch was achieving higher accuracy then the earlier one.

As the patients will upload MRI scans and health reports, it will be added to the database and which will help the machine learning model to achieve greater accuracy and can be accessed any time by the patients.

Contents of the Project:

Root Project directory name : **web**
Contents of web directory:

- **database_dump** : This directory contains the database dump required for the functioning of the website.
- **website**: This folder contains the frontend, backend and the machine learning model needed for the website.
- **config.json**: This is a configuration file for the database URI.
- **requirements.txt**: This file contains all the python libraries required for the server to operate.
- **run.py**: This python file needs to be run to start the website.

Steps to host the project to localhost:

- Open the terminal on the **web** directory and create a virtual environment then activate the environment and install all the libraries using requirements.txt file. [**Note : Do not close the terminal**]

  Ex: (For Windows)

  ```
  py -m venv env
  .\env\Scripts\activate
  pip install -r requirements.txt
  ```
- Start Apache and Mysql from Xampp or any other web server solution stack. Then open the phpMyadmin from localhost and create a database named “**inferno**” or the name can be customised from **config.json** file.
- Import the database dump available in the database_dump folder using phpMyadmin GUI in localhost.
- Run the following command in the terminal:

  ```
  python run.py
  ```

  The server will be up and running on:
  http://127.0.0.1:5000/