Brain Guardian

Mid-semester Progress Report

Submitted By

Adarsh Kumar (B520003)
Dheeraj Singh (B520019)
Nirmit Gupta (B520034)

Under the guidance of

Dr. Hemanta Kumar Pati
Assistant Professor in Computer Science and Engineering Department

Department of Computer Engineering
International Institute of Information Technology, Bhubaneswar
February, 2024

Abstract

In the healthcare industry, the lack of a unified system creates a significant challenge in integrating brain health management seamlessly. This fragmentation prevents accurate prediction of brain tumors, efficient scheduling of specialist appointments, and makes it difficult to facilitate video consultations with healthcare professionals. This disconnection hinders patient access to comprehensive brain health services, leading to delayed diagnoses and suboptimal treatment outcomes.

Introducing **BrainGuardian** - a revolutionary project that offers a Comprehensive Brain Health Management System. This cutting-edge solution combines advanced machine learning algorithms for predictive analysis, an easy-to-use appointment booking interface, and a secure video consultation platform. The primary goal is to transform the way we approach brain health by providing a unified and accessible solution for both patients and healthcare providers.

Aiming to improve early detection capabilities and increase the chances of successful intervention, a predictive model for brain tumors is being developed. Additionally, an efficient appointment booking system tailored specifically to brain health is being created to reduce waiting times and enhance the patient experience. To further improve access to healthcare services, a robust video consultation platform is being implemented, allowing patients to consult with healthcare professionals remotely and overcome geographical constraints.

BrainGuardian promises better patient outcomes, early detection of brain health issues, and increased efficiency in neurology healthcare services. This report explains the project's methodology, system architecture, development processes, and expected impact on healthcare. By combining technology and healthcare expertise, **BrainGuardian** represents a significant leap forward in patient care.

Contents

1 Introduction

A brain tumor refers to an abnormal collection or mass of cells within the brain. The skull, which encloses the brain, has limited space, and any growth within this confined area can lead to complications. Brain tumors can be either **cancerous (malignant)** or **noncancerous (benign)**. As benign or malignant tumors grow, they can increase the pressure inside the skull. This elevated pressure can cause brain damage and pose a life-threatening risk.The early detection and classification of brain tumors are crucial areas of research in medical imaging. Accurate classification aids in selecting the most suitable treatment method, potentially saving patients' lives.Malignant tumors can expand quickly and disperse across the surrounding brain tissue, whereas benign tumors tend to grow slowly. However, benign tumors can also be dangerous as their proliferation may affect surrounding brain tissues. **About 70% of the tumors are benign, and 30% are malignant**. So far, **more than 120 different brain tumors** including **meningioma, glioma, and pituitary** as the most popular ones have been detected and identified. Among these three, meningioma tumors are perhaps the most prominent primary brain tumor in the meninges and affect the brain and spinal cord. On the other hand, glioma tumors grow from glial cells called astrocytes. The most prominent tumor of glioma is an astrocytoma, a low-risk tumor that suggests slow development. However, high-risk glioma is one of the most severe brain tumors. Pituitary is another type of tumor that is due to excessive growth of brain cells in the pituitary gland of the brain. Therefore, early diagnosis of a brain tumor is essential due to its deadly aspect.

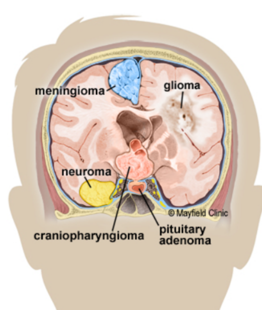


Figure 1: Brain tumors may grow from nerves (neuroma), dura (meningioma), or pituitary gland (craniopharyngioma or pituitary adenoma). They may also grow from the brain tissue itself (glioma). As they grow they may compress normal tissue and cause symptoms.

Besides this, in today's fast-paced world, accessing specialized medical care can present formidable challenges, particularly for individuals residing in remote areas or confronting mobility limitations. The barriers to accessing quality healthcare are multifaceted, often compounded by geographic distance, limited transportation infrastructure, and socioeconomic disparities. For those living in rural or isolated regions, the nearest medical facility equipped with specialized expertise may be hours away, necessitating arduous and costly journeys. Furthermore, individuals with mobility impairments or chronic health conditions may encounter additional hurdles in physically reaching healthcare facilities, further exacerbating their struggle to obtain timely and essential medical attention.For this a **website that aims to bridge this gap by offering a range of services tailored to meet the needs of individuals seeking diagnosis, treatment, and support for brain tumors is needed.**

1.1 Objectives

- **Enhancing Diagnostic Capabilities**: Our primary objective is to equip healthcare professionals with robust software tools for accurate tumor identification and comprehensive understanding of their underlying causes. By providing doctors with sophisticated diagnostic capabilities, we empower them to deliver superior patient care and treatment outcomes.

- **Time Optimization and Efficiency**: Time is of the essence in healthcare, particularly in cases of critical illness such as tumors. Our application aims to streamline the diagnostic process, saving valuable time for both healthcare providers and patients. By expediting the identification and diagnosis of tumors, we enable timely intervention and treatment initiation, ultimately improving patient outcomes.

- **Early Intervention and Timely Consultation**: Early detection is paramount in the management of tumors. Our application facilitates prompt identification of tumors at their incipient stages, allowing for timely intervention and appropriate treatment strategies. Additionally, it provides a platform for patients to seek timely consultation and expert medical advice, ensuring proactive management of their health concerns.

2 Literature Survey

The paper titled **"Accurate Brain Tumor Detection Using Deep Convolutional Neural Network"** presents a novel approach to brain tumor detection by integrating the "23-layers CNN" framework with the transfer learning-based VGG16 model. The study demonstrates that this fusion technique yields impressive results, achieving accurate detection of brain tumors in test images. Notably, the proposed method showcases exceptional performance without encountering overfitting issues, indicating its robustness and potential for clinical application. By leveraging the strengths of both frameworks, the approach offers a promising solution for enhancing the accuracy and reliability of brain tumor detection using deep learning methodologies.

The paper titled **"Identification of Brain Tumor using Image Processing Techniques"** authored by Praveen Gamage and published on ResearchGate in September 11, 2017, provides a comprehensive survey of methods for identifying brain tumors through MRI images. The survey categorizes the techniques into four main sections: pre-processing, image segmentation, feature extraction, and image classification.

The study titled "**A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images**" presents a novel hybrid model combining convolutional neural networks (CNN) and support vector machines (SVM) for tumor detection and classification in MRI brain images. The hybrid model achieved an impressive accuracy of 98.4959%, outperforming both SVM (72.5536%) and CNN (97.4394%) when used separately. This suggests that the proposed hybrid approach offers more effective and improved techniques for tumor classification. The study also suggests the potential for further enhancement by incorporating faster CNN models with SVM and exploring optimization methods such as bio-inspired algorithms.

3 Problem statement and Solution Approach

In the contemporary healthcare landscape, a pervasive challenge exists in establishing a seamless and integrated system for managing brain health. **The absence of a cohesive platform impedes the accurate prediction of brain tumours, disrupts the efficient scheduling**

of appointments with specialists, and inhibits the facilitation of video consultations with healthcare professionals. Patients, as a result, encounter difficulties in accessing comprehensive brain health services, leading to delayed diagnoses and suboptimal treatment outcomes. This project seeks to address the following pivotal issues:

- **Absence of Predictive Tools for Brain Tumors:** The dearth of sophisticated and user-centric predictive tools for the early detection of brain tumours is a significant hindrance to timely medical intervention. The development of a precise predictive model is imperative to empower healthcare providers in identifying potential brain tumours in their nascent stages, thereby augmenting the likelihood of successful treatment.

- **Inefficiencies in Appointment Booking Systems:** Current appointment scheduling systems within healthcare institutions lack user-friendliness and fail to provide patients with convenient and timely access to specialised services. The establishment of an efficient appointment booking system, tailored specifically for brain health, is integral to streamlining the process, reducing waiting times, and enhancing the overall patient experience.

- **Restricted Access to Remote Consultations:** Geographical constraints and the absence of a robust telehealth infrastructure contribute to challenges in accessing brain health consultations. The implementation of a secure and comprehensive video consultation platform aims to overcome these barriers, enabling patients to remotely consult with healthcare professionals, receive expert advice, and seamlessly follow up on their treatment plans.

  The proposed solution entails the development of an integrated health management system that integrates sophisticated deep learning algorithms for brain tumor prediction, an intuitive appointment booking interface, and a secure video consultation platform. This holistic system aims to redefine the paradigm of brain health management, providing a comprehensive and accessible solution for both patients and healthcare providers.

3.1 Workflow

3.1.1 ML & DL Workflow

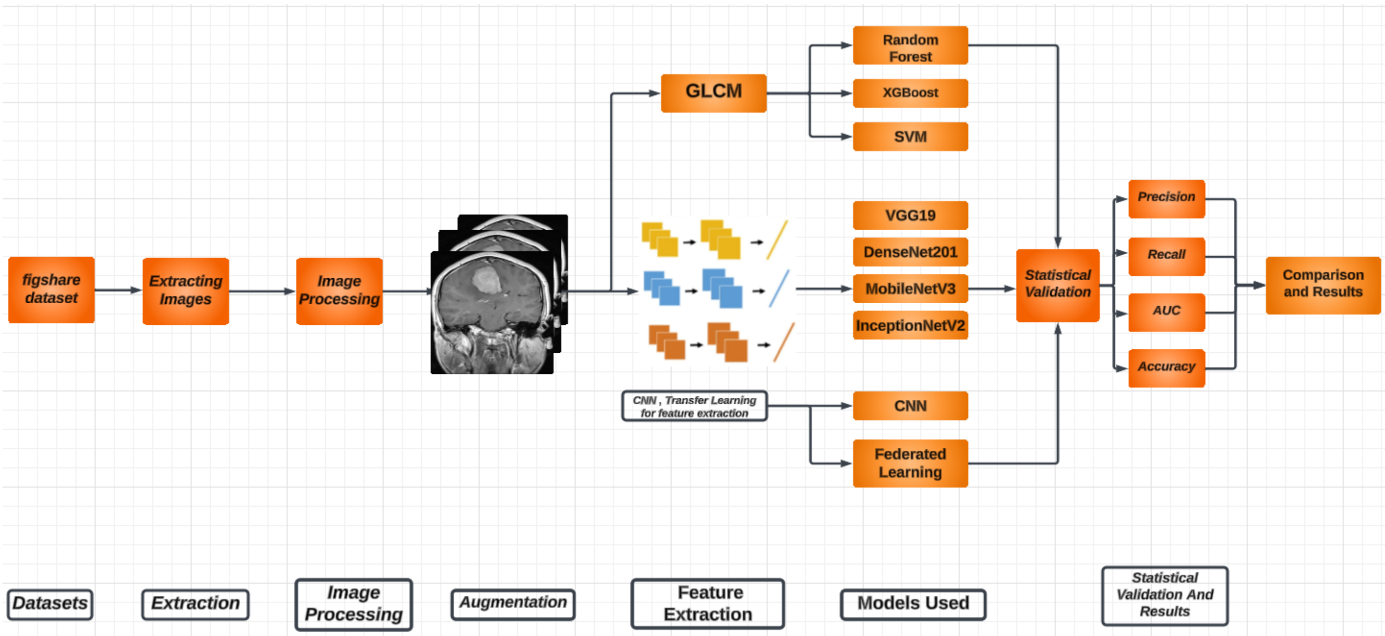


Figure 2: ML & DL Workflow.

Dataset

This dataset is a combination of the following **three datasets** :

- figshare

The figshare dataset consists of 3064 T1-weighted images which were taken from 233 patients.The images were classified into three types of tumor - Meningioma (708 slices) , Pituitary (930 slices) and last Glioma(1426 slices). The images were stored in matlab data format in a struct format.

Image Processing

Firstly the images were extracted from .mat files and stores as .png files. When it is preprocessing of images,so the images taken from fighsare website were having dimensions 512x512x3. The images had many black bankground area. So for preprocessing we took a code from Kaggle where firstly all the images are converted to grayscale.Then thresholding is applied to separate background and foreground.After this various erosions and dilations .Further took the largest contour from the image and made a new image from the extreme points of this contour.The final image size was 256x256x3.Finally the new image was ready for analysis.

Augmentation

We ensure uniformity and quality in training data by rescaling pixel values, using data augmentation techniques, and standardizing image size. This not only improves the model's ability to learn from data, but also improves its ability and overall ability regarding invisible images. In general, preprocessing plays an important role in improving model performance, reducing overfitting, and effectively utilizing image data in many applications. Using Keras's ImageDataGenerator, this script uses advanced techniques including retouching and data enhancement (such as cropping and scale transformations). It then creates bulk augmented and normalized data from the input DataFrame, which contains stored data and class labels for training, validation, and testing data. This simplification increases the efficiency of image data for training, thus improving the learning ability of the model and generalization of the data.

Feature Extraction

In tasks involving image processing and computer vision, feature extraction is essential. Large volumes of data are included in images, and it is crucial for many applications to extract useful information from them. Feature extraction here is done using two methods -

- CNN
- GLCM

Feature extraction from convolutional neural networks (CNN) is a simple technology in computer vision and deep learning. CNN is good at learning hierarchical representation of input images and capturing features at different levels of abstraction. Feature extraction involves passing the input image through a series of CNN preprocessing methods (such as VGG, ResNet, or Inception) and extracting the function from one or more intermediate layers. These functions represent high-level features learned from the model, such as edges, textures, shapes, and residuals. Using representations learned from CNNs, we can extract landmarks from input images, which can be used for the following tasks, including image classification, detection products, and return images. This approach is especially important when dealing with limited data or optimizing a predefined model for a specific task. Overall, feature extraction from CNNs provides a powerful and effective way to extract rich images and unlock the potential of deep learning in computer vision applications.

GLCM stands for Grey Level Co-occurence Matrix.It considers relationships between neighbouring pixels,providing rich texture information. The main idea behind GLCM is to examine how frequently adjacent pixels in an image—that is, gray levels or intensities—occur together.

Pixel offsets and directions: The spatial organization of textures is captured by GLCM by counting co-occurrences in a given pixel offset (often 1 or 2) and in a given direction (e.g., horizontal, vertical, diagonal).

Thus, GLCM is calculated using the following three parameters:

1. Distance (d): The displacement between two pixels is called distance (d).

2. Angle (): Usually in 0°, 45°, 90°, and 135°, this is the direction in which pixel pairs are evaluated.

3. Number of Gray Levels (G): The image's total number of distinct intensity levels.

Classification

After extracting features from various previously learned convolutional neural network (CNN) models and GLCM, we use machine learning (ML) algorithms and deep learning classifiers for classification. The extracted features are used as a high-level representation of the image input, capturing important patterns and features. We use feature extracted via GLCM to train classes that can predict unseen images using ML algorithms such as Support Vector Machines (SVM), Random Forest (RNF), and XGBoost. Additionally, I trained a custom CNN architecture for training also.

Transfer Learning :- Reusing a pre-trained model on a new task is known as transfer learning in machine learning. A machine uses transfer learning to increase its generalization about another task by using the knowledge it has learned from a prior one.While there are many advantages to transfer learning, the three primary ones are reduced training time, improved neural network performance (for the most part), and reduced data requirements.

To train a neural network from beginning, a large amount of data is typically required, but access to that data isn't always possible. This is when transfer learning comes in useful. Because the model has previously been pre-trained, transfer learning allows for the construction of strong machine learning models with relatively minimal training data.

Federated Learning :- Federated learning is a decentralized method of training machine learning models (often called collaborative learning). Data exchange from client devices to global servers is not necessary. Rather, the model is trained locally using the raw data on edge devices, hence improving data privacy. The local modifications are aggregated to build the final model in a shared manner.Federated Learning provides two major benefits-

- **Privacy**-Federated learning allows training to happen locally on the edge device, avoiding potential data breaches, in contrast to traditional approaches that send data to a central server for training.

- **Heterogenity**-Access to heterogeneous data is ensured via federated learning, which makes data dispersed across various organizations, places, and devices accessible.

Statistical Validations

After classification we used statistical analysis methods to evaluate the effectiveness and robustness of classification models in my deep learning activities. These metrics such as precision, accuracy, recall, and F1 score are evaluated on test data that differs from the reported data. By evaluating these metrics, one gains a better understanding of the model's ability to generalize to unobserved data and make accurate predictions across different groups.This rigorous analysis not only provides a better understanding of the model's strengths and limitations, but also informs decisions about model selection, hyperparameter tuning, and potential development availability for pipelines. In general, statistical validation plays an important role in the validity and reliability of deep learning models and ultimately contributes to their successful use in the world of applications.

3.1.2 Website Workflow

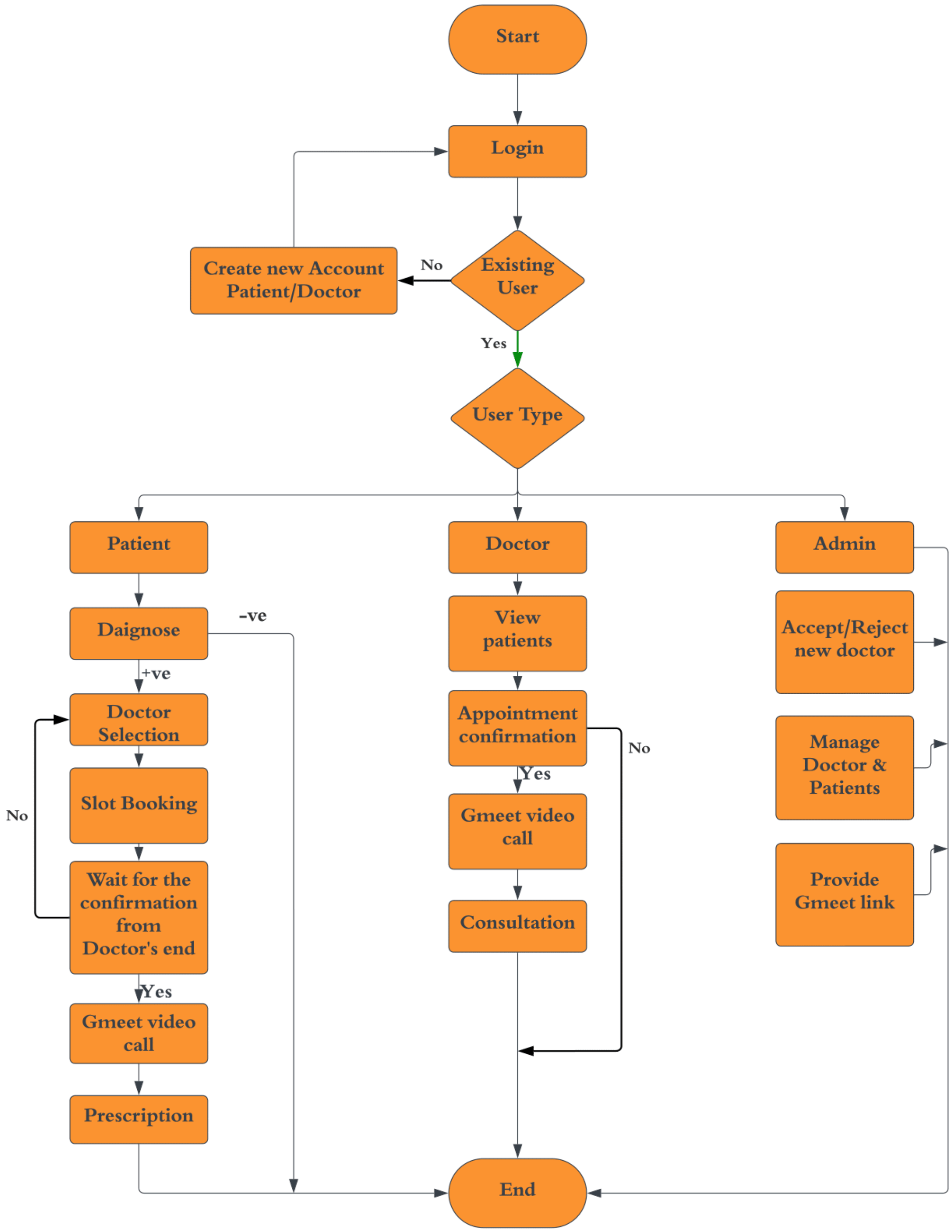


Figure 3: Website Workflow.

Patient's perspective

To begin, the patient will log in to the system. If a new user is created, they will create a new account, or if an existing user is created, they will log in with their existing credentials. Once logged in, selecting "patient" will take them to a page where they can diagnose their MRI reports, book appointments, and manage their account. To book an appointment, the patient will choose a doctor by name or specialty and view their available appointment slots. Once a slot is selected and booked, the patient will receive a confirmation email or SMS. On the day of the appointment, the patient will wait for the doctor to confirm the appointment before joining a video call. After the consultation, the doctor may prescribe medication, and the patient can end the call.

Doctor's perspective

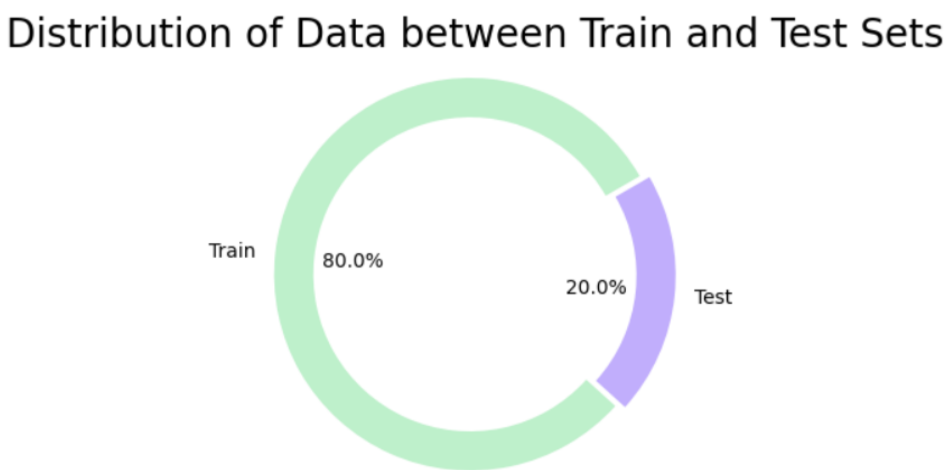
To access the system, doctors can log in with their existing accounts. Once logged in, doctors can conveniently view a list of their patients and confirm or reject appointment requests as needed. After confirming an appointment, doctors can easily select a time slot for the consultation and conduct a video call using Google Meet. Finally, following the consultation, doctors can create a prescription for their patients to ensure they receive the necessary care.

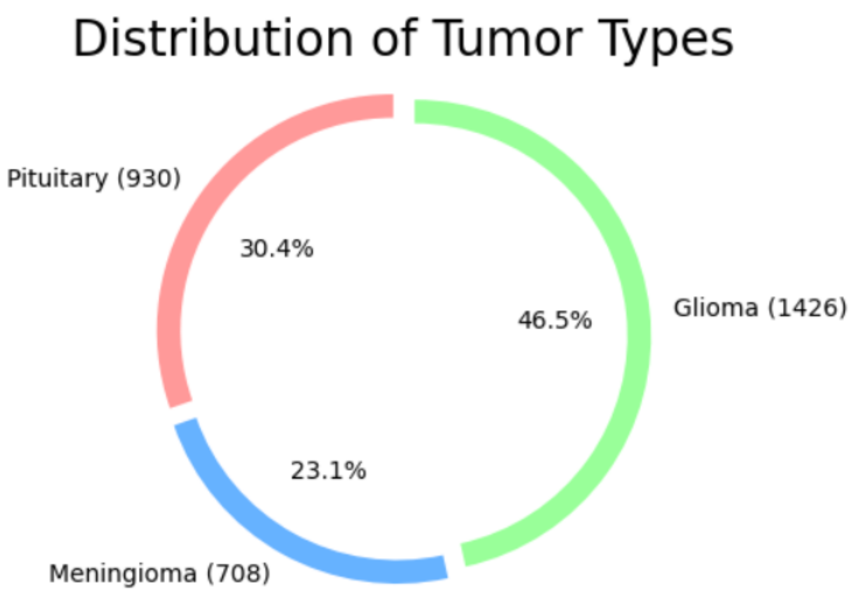
Admin's perspective

The process initiates as the administrator logs into the platform. Once logged in, the administrator has the ability to manage doctors and patients, which encompasses viewing their details, approving or rejecting new doctor registrations, and overseeing their appointments. Additionally, the administrator can review and handle appointment confirmations as needed.

4 Results and Discussion

- **During model preparation, dataset distribution was as follows.**





4.0.1 Transfer Learning

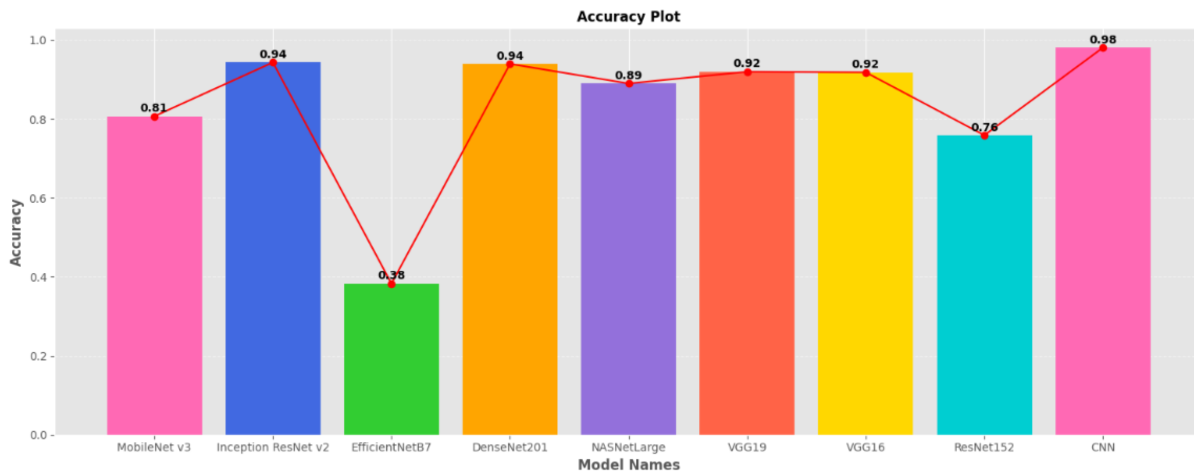
After successful implementation of workflows,following results were obtained.

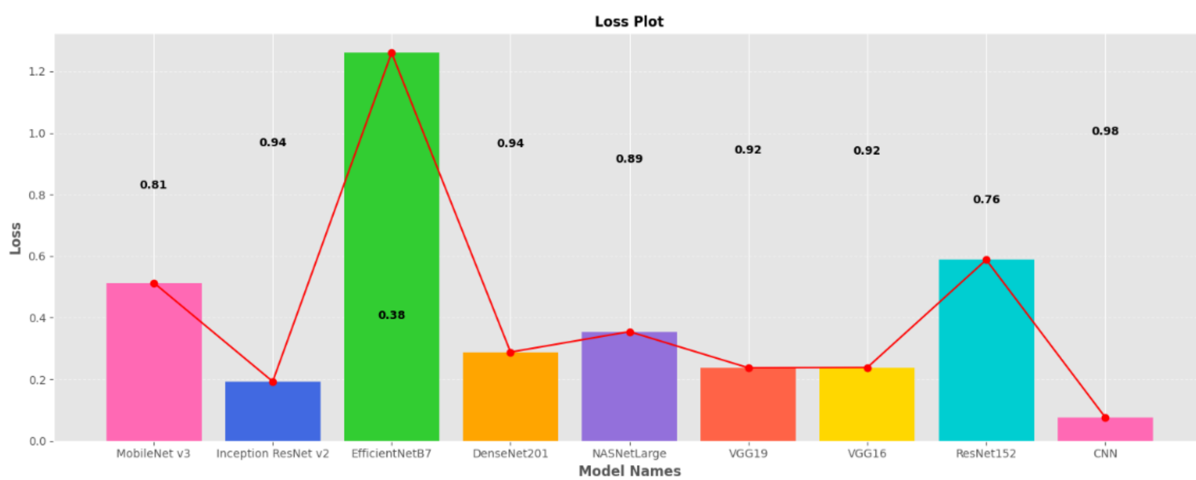
Models Trained

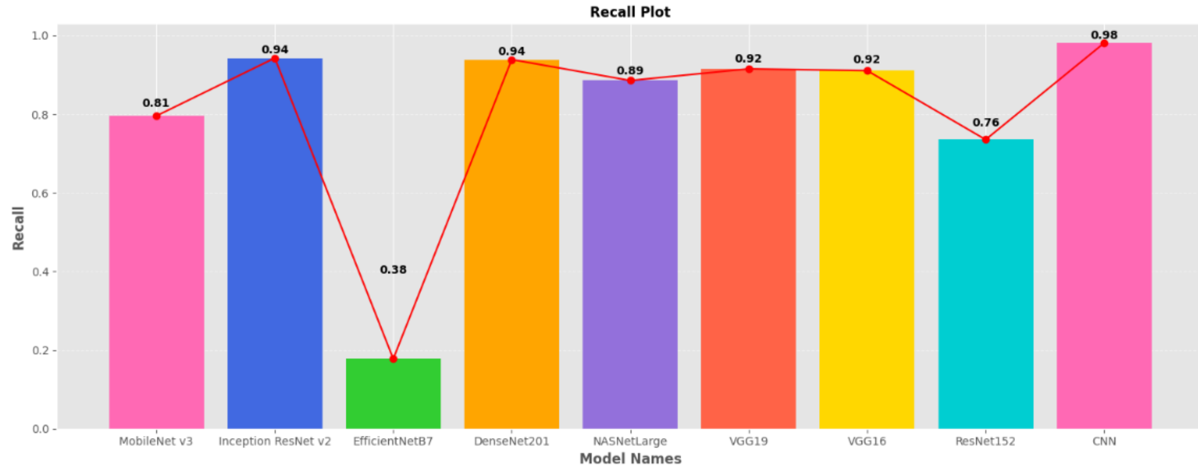
- *MobileNetV3*
- *InceptionResNetV2*
- *DenseNet201*

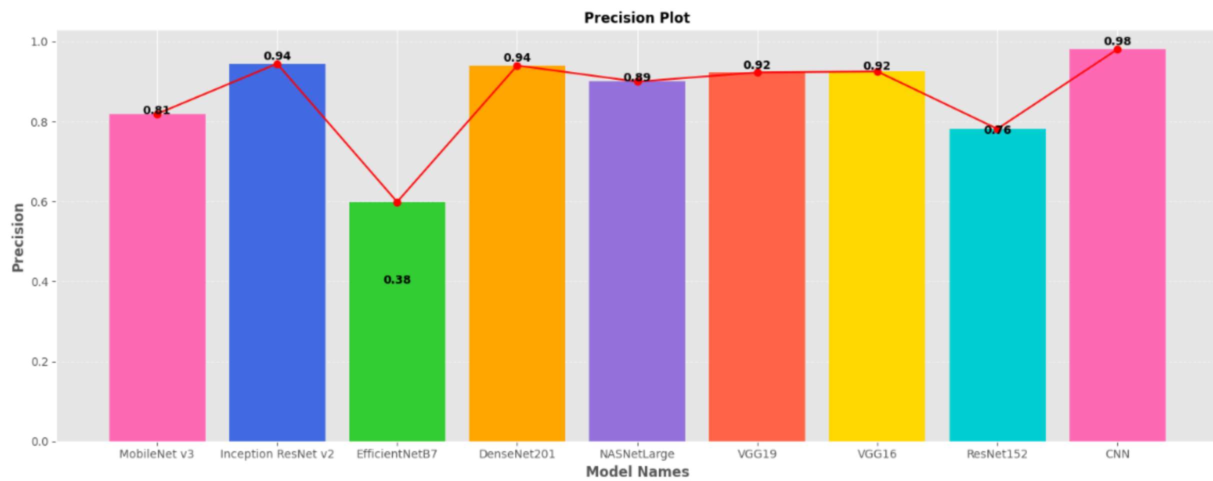
- *VGG19*
- *Custom CNN Model*
- *Federated Learning*
- *GLCM with XGBoost,Random Forest,SVM*

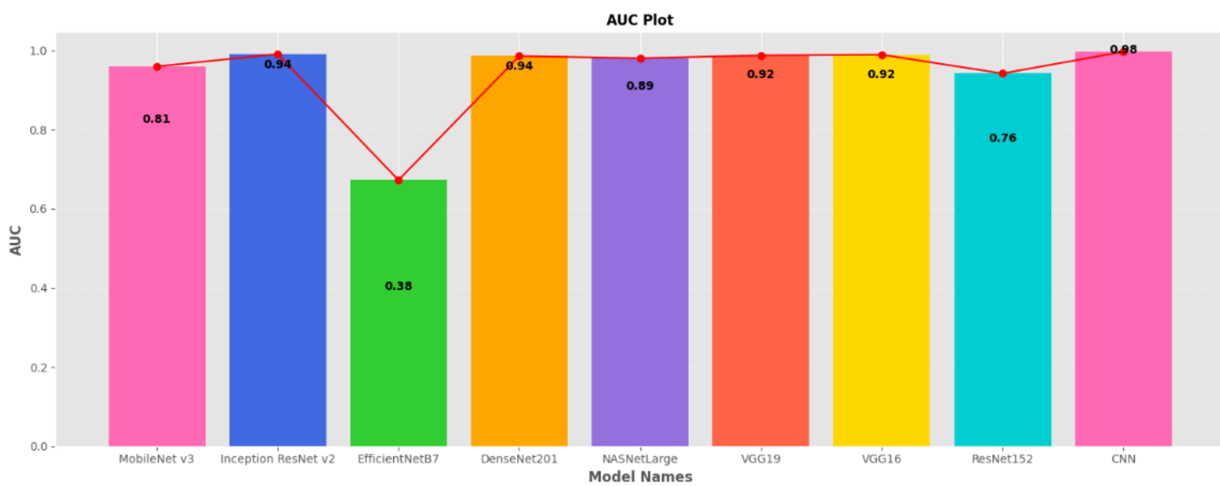
Plots






Recall Plot
Recall
0.81
0.94
0.38
0.94
0.89
0.92
0.92
0.76
0.98
MobileNet v3
Inception ResNet v2
EfficientNetB7
DenseNet201
NASNetLarge
VGG19
VGG16
ResNet152
CNN
Model Names


Precision Plot
Precision
0.81
0.94
0.38
0.94
0.89
0.92
0.92
0.76
0.98
MobileNet v3
Inception ResNet v2
EfficientNetB7
DenseNet201
NASNetLarge
VGG19
VGG16
ResNet152
CNN
Model Names


AUC Plot
AUC
0.81
0.94
0.38
0.94
0.89
0.92
0.92
0.76
0.98
MobileNet v3
Inception ResNet v2
EfficientNetB7
DenseNet201
NASNetLarge
VGG19
VGG16
ResNet152
CNN
Model Names

Performance Metrics

To evaluate the performance of various architectures and compare our results we use different evaluation metrics including, accuracy, precision, recall, false-positive rate (FPR), true negative rate (TNR), and F1-score. These metrics are calculate as follows:

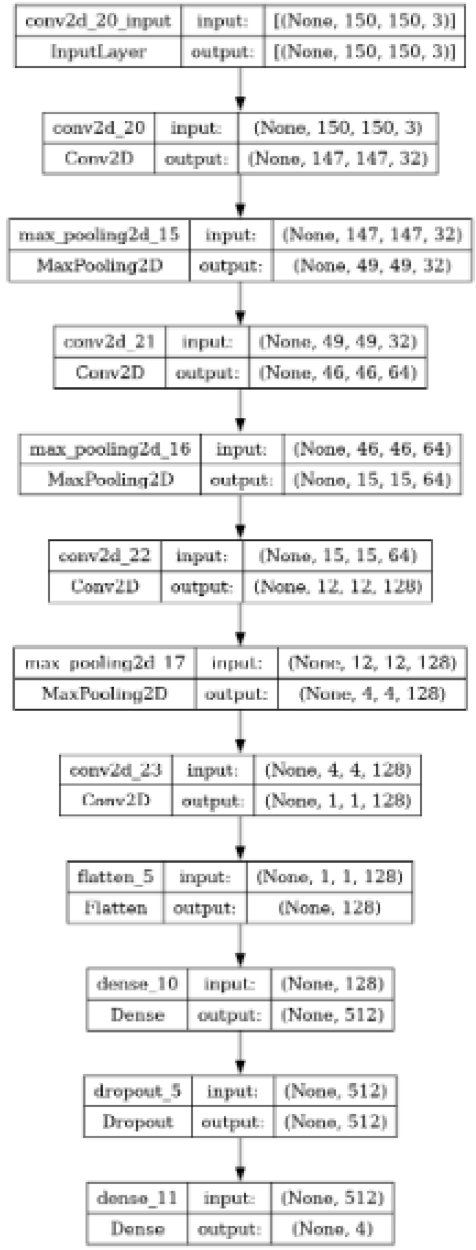
- **Accuracy-** $\frac{TP+TN}{TP+FP+TN+FN}$
- **Recall-** $\frac{TP}{TP+FN}$
- **Precision-** $\frac{TP}{TP+FP}$
- **F1 Score-** $2 * \frac{Recall*Precision}{Recall+Precision}$

Where TP stands for true positive, FP stands for false positive, TN stands for true negative, and FN stands for false negative.

It is clear that the proposed **CNN architecture outperforms** other models in evaluating various deep learning models for task classification, such as MobileNetv3, InceptionResNetv2, EfficientNetB7, DenseNet201, NASNetLarge, VGG19, VGG16, ResNet152, and custom CNN architectures on many key performance indicators. With low loss and high accuracy, the CNN architecture demonstrates robustness and efficiency in image classification. Moreover, its high precision and recall scores show that it can reduce false positives and false negatives, which is important in clinical applications where classification errors cause serious consequences. Additionally, the CNN architecture exhibits excellent AUC scores, indicating its ability to discriminate between classes. These findings demonstrate the superiority of CNN models in classification of tasks, diagnostic ability, and other important areas where accuracy and reliability are important.

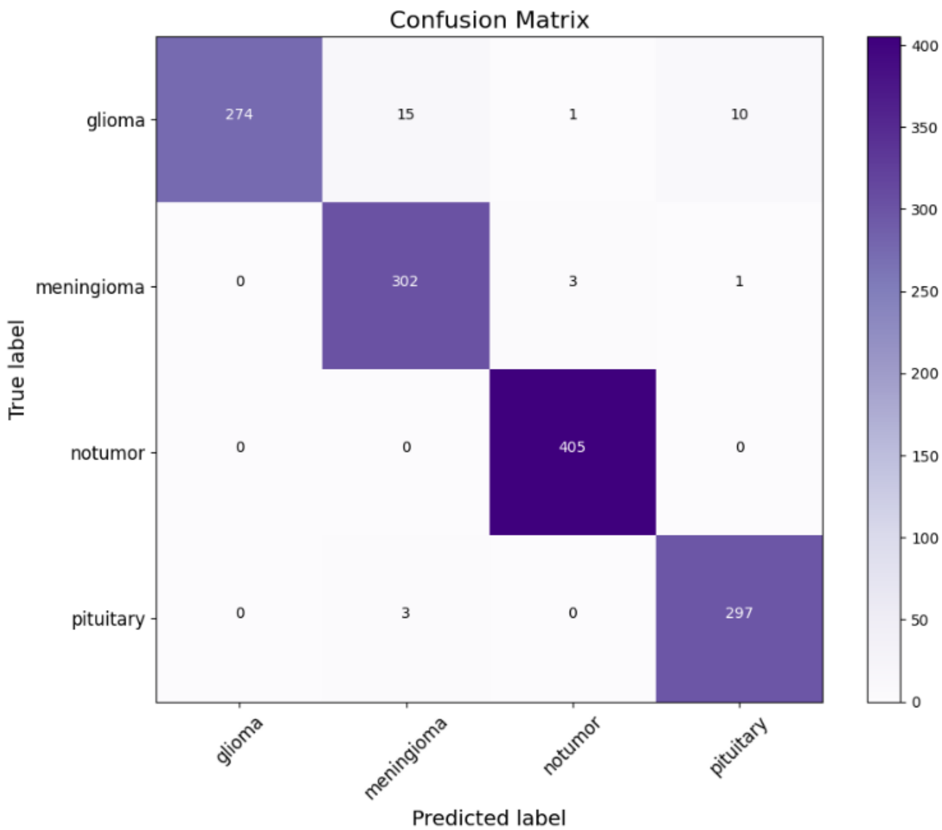
The CNN architecture was as follows-

This architecture consists of four layers, each combined with a max-pooling layer to eliminate critical features while reducing dimensions. After the convolution layer, the feature map is flattened



for two full layers. The first layer consists of 512 units enabled by linear tuning (ReLU), followed by the output method to reduce the overfitting problem. The second layer then uses the softmax activation function to activate the resulting class. The model was carefully compiled using the Adam optimization tool and the categorical cross-entropy loss function and evaluated on metrics such as accuracy, AUC, precision, and recall, allowing a complete evaluation of its entire distribution.

Confusion Matrix -



Class 0 (glioma): 99% precision indicates that 99% of the instances classified as glioma were correctly predicted, with a 1% chance of misclassifying other types as glioma. 96% recall suggests that 96% of actual glioma instances were correctly classified, with a 4% chance of missing glioma cases.

Class 1 (meningioma): 96% precision implies that 96% of instances classified as meningioma were correct, with a 4% chance of misclassifying other types as meningioma. 98% recall indicates that 98% of actual meningioma instances were correctly classified, with a 2% chance of missing meningioma cases.

Class 2 (tumor): 99% precision indicates that 99% of instances classified as tumor were correct, with a 1% chance of misclassifying other types as tumor. 99% recall suggests that 99% of actual tumor instances were correctly classified, with a 1% chance of missing tumor cases.

Class 3 (pituitary): 99% precision implies that 99% of instances classified as pituitary were correct, with a 1% chance of misclassifying other types as pituitary. 99% recall indicates that 99% of actual pituitary instances were correctly classified, with a 1% chance of missing pituitary cases.

Overall, the confusion matrix demonstrates high precision and recall values across all classes, indicating strong performance in accurately classifying brain tumor types. The model shows minimal confusion between classes, with most instances correctly classified.

4.0.2 Hybrid Learning

Model	Classifier	Accuracy
VGG19	XGB	25.6%
VGG19	GNB	30.5%
VGG19	SVC	19.9%
CNN	XGB	25.0%
CNN	GNB	25.0%
CNN	SCV	22.8%

Table 1:

This table shows the accuracy scores for different classification models used for two different neural network architectures (VGG19 and CNN). Among the models tested, Gaussian Naive Bayes (GNB) achieved the highest accuracy of 30.59% when used with the VGG19 architecture. This shows that GNB, a simple probabilistic classifier, is an effective classifier for VGG19 features in this case. **However, the overall performance of the classifiers across all combinations is still low, with accuracy scores around 20% to 30%. This situation shows the complexity and problems in classifying products using models and designs**.

- **During development of the Website**

4.0.3 Model Schema Creation

The MERN stack project utilizes MongoDB with Mongoose for robust data modeling. Three schemas—user, doctor, and appointment—are pivotal components of the system's architecture. The user schema defines user data storage, including basic details like name, email, password, and optional attributes such as age, gender, and profile picture. Additionally, isAdmin and isDoctor fields facilitate role management. The doctor schema, linked to users via userId, stores specialized information like experience, fees, and timestamps for updates. Meanwhile, the appointment schema manages scheduling, connecting users and doctors via userId and doctorId. It features fields for date, time, and status, ensuring effective appointment management. Through meticulous schema design and Mongoose's capabilities, the project ensures seamless data interaction and scalability. The schemas enable comprehensive user and doctor profiling, fostering transparency and trust within the platform. Appointment management is streamlined, enhancing user experience and facilitating efficient healthcare service delivery. The project's reliance on Mongoose underscores its commitment to robust data modeling and system scalability, laying a solid foundation for future enhancements.

In summary, the MERN stack project employs MongoDB and Mongoose to craft intricate user, doctor, and appointment schemas, optimizing data organization and retrieval. This meticulous design enhances user experience through transparent profile management and streamlined appointment scheduling. Leveraging Mongoose not only ensures scalability but also sets a solid foundation for future system enhancements and expansions. The project's architecture underscores appointmentModel.js > ...

Adarsh-k0802, last week | 1 author (Adarsh-k0802)

```javascript
const mongoose = require("mongoose");

const schema = mongoose.Schema(
  {
    userId: {
      type: mongoose.SchemaTypes.ObjectId,
      ref: "User",
      required: true,
    },
    doctorId: {
      type: mongoose.SchemaTypes.ObjectId,
      ref: "User",
      required: true,
    },
    date: {
      type: String,
      required: true,
    },
    time: {
      type: String,
      required: true,
    },
    status: {
      type: String,
      default: "Pending",
    },
  },
  {
    timestamps: true,
  }
);
```

doctorModel.js > ...

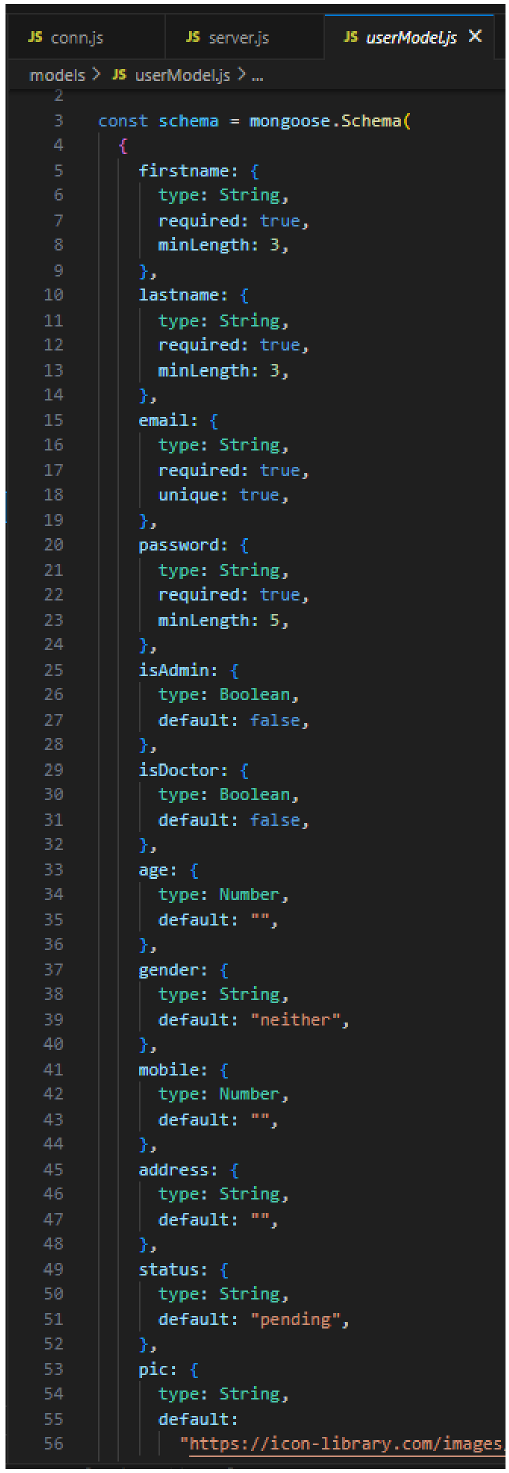
Adarsh-k0802, last week | 1 author (Adarsh-k0802)

```javascript
const mongoose = require("mongoose");

const schema = mongoose.Schema(
  {
    userId: {
      type: mongoose.SchemaTypes.ObjectId,
      ref: "User",
      required: true,
    },
    specialization: {
      type: String,
      required: true,
    },
    experience: {
      type: Number,
      required: true,
    },
    fees: {
      type: Number,
      required: true,
    },
    isDoctor: {
      type: Boolean,
      default: false,
    },
  },
  {
    timestamps: true,
  }
);
```

Figure 4: Doctor Model & Appointment Model

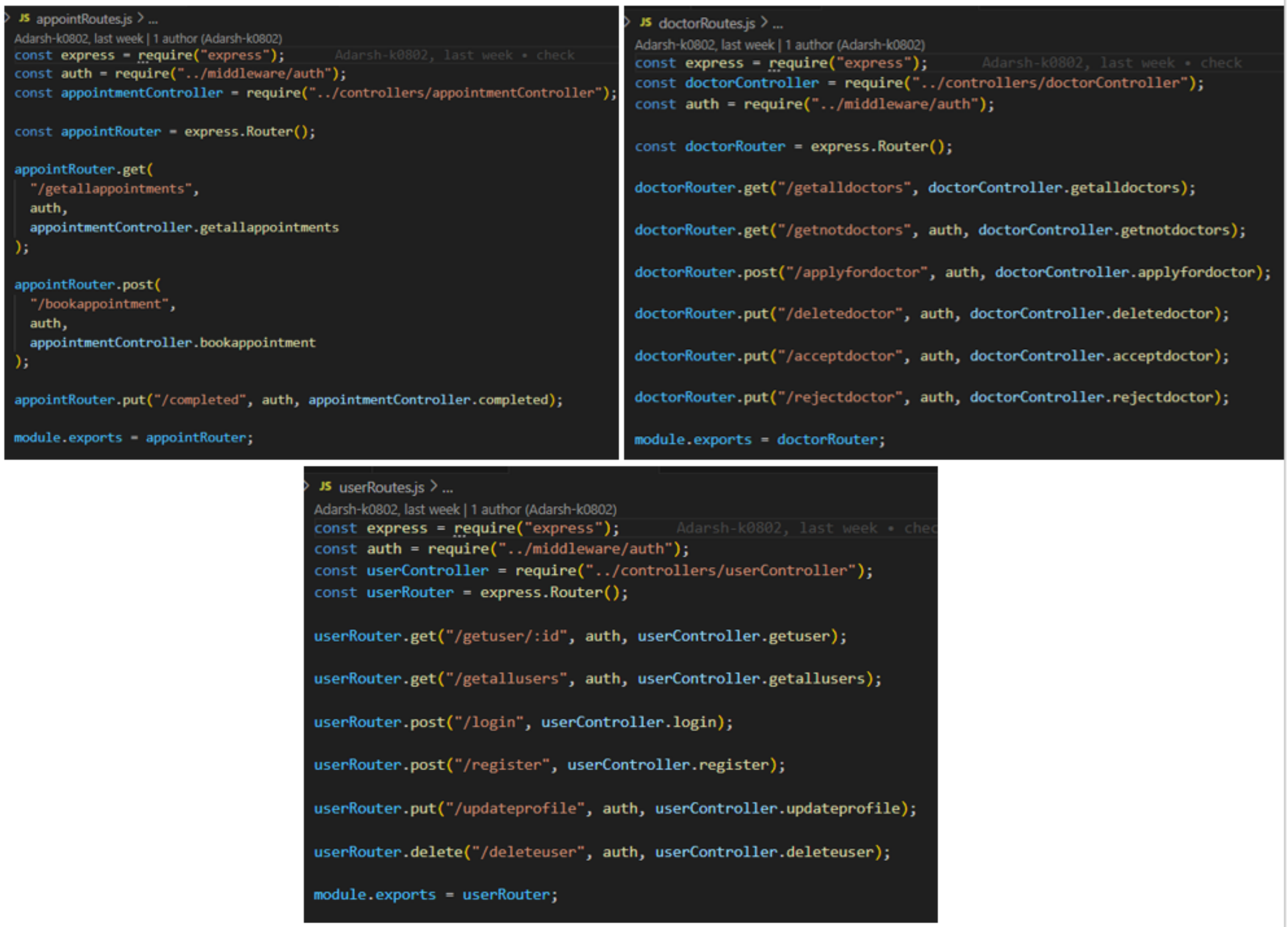
its commitment to robust data modeling and seamless interaction, fostering trust and efficiency within the platform.




```
const schema = mongoose.Schema(
  {
    firstname: {
      type: String,
      required: true,
      minLength: 3,
    },
    lastname: {
      type: String,
      required: true,
      minLength: 3,
    },
    email: {
      type: String,
      required: true,
      unique: true,
    },
    password: {
      type: String,
      required: true,
      minLength: 5,
    },
    isAdmin: {
      type: Boolean,
      default: false,
    },
    isDoctor: {
      type: Boolean,
      default: false,
    },
    age: {
      type: Number,
      default: "",
    },
    gender: {
      type: String,
      default: "neither",
    },
    mobile: {
      type: Number,
      default: "",
    },
    address: {
      type: String,
      default: "",
    },
    status: {
      type: String,
      default: "pending",
    },
    pic: {
      type: String,
      default:
        "https://icon-library.com/images
```

Figure 5: User Model

4.0.4 API Routes Creation



JS appointRoutes.js > ...
Adarsh-k0802, last week | 1 author (Adarsh-k0802)

```javascript
const express = require("express");        Adarsh-k0802, last week • check
const auth = require("../middleware/auth");
const appointmentController = require("../controllers/appointmentController");

const appointRouter = express.Router();

appointRouter.get(
  "/getallappointments",
  auth,
  appointmentController.getallappointments
);

appointRouter.post(
  "/bookappointment",
  auth,
  appointmentController.bookappointment
);

appointRouter.put("/completed", auth, appointmentController.completed);

module.exports = appointRouter;
```

JS doctorRoutes.js > ...
Adarsh-k0802, last week | 1 author (Adarsh-k0802)

```javascript
const express = require("express");        Adarsh-k0802, last week • check
const doctorController = require("../controllers/doctorController");
const auth = require("../middleware/auth");

const doctorRouter = express.Router();

doctorRouter.get("/getalldoctors", doctorController.getalldoctors);

doctorRouter.get("/getnotdoctors", auth, doctorController.getnotdoctors);

doctorRouter.post("/applyfordoctor", auth, doctorController.applyfordoctor);

doctorRouter.put("/deletedoctor", auth, doctorController.deletedoctor);

doctorRouter.put("/acceptdoctor", auth, doctorController.acceptdoctor);

doctorRouter.put("/rejectdoctor", auth, doctorController.rejectdoctor);

module.exports = doctorRouter;
```

JS userRoutes.js > ...
Adarsh-k0802, last week | 1 author (Adarsh-k0802)

```javascript
const express = require("express");        Adarsh-k0802, last week • chec
const auth = require("../middleware/auth");
const userController = require("../controllers/userController");
const userRouter = express.Router();

userRouter.get("/getuser/:id", auth, userController.getuser);

userRouter.get("/getallusers", auth, userController.getallusers);

userRouter.post("/login", userController.login);

userRouter.post("/register", userController.register);

userRouter.put("/updateprofile", auth, userController.updateprofile);

userRouter.delete("/deleteuser", auth, userController.deleteuser);

module.exports = userRouter;
```

Figure 6: Appointment Routes, Doctor Routes, User Routes

The **userRoutes.js** file orchestrates various functionalities pertaining to user management on the website. Each route serves a distinct purpose:

/getuser/:id: Retrieves information about a specific user based on their ID. Utilizes userController.getuser to fetch data from the database and return it in the response.

/getallusers: Fetches a list of all users registered on the platform. Calls userController.getallusers to retrieve user data and send it back in the response.

/login: Allows users to log in using their credentials. Executes userController.login to validate the username and password against the database and issues a token upon successful authentication.

/register: Enables new user registration. Utilizes userController.register to create a new user in the database with provided credentials and issues a token upon successful registration.

/updateprofile: Facilitates updating a user's profile information. Requires authentication and utilizes userController.updateprofile to modify user data in the database.

/deleteuser: Allows users to delete their accounts. Requires authentication and invokes userController.deleteuser to remove the user's data from the database.

The **doctorRoutes.js** file manages functionalities related to doctors on the website. Here's an overview:

/getalldoctors: Retrieves information about all registered doctors. Likely calls doctorController.getalldoctors to fetch doctor data from the database.

/getnotdoctors: Fetches information about users who haven't yet registered as doctors. Requires authorization and probably calls doctorController.getnotdoctors to retrieve and return the data.

/applyfordoctor: Allows users to apply for the role of a doctor. Requires authorization and likely invokes doctorController.applyfordoctor to process the application.

/deletedoctor: Enables deleting doctor profiles. Requires authorization and presumably calls doctorController.deletedoctor to remove the specified doctor's data from the database.

/acceptdoctor: Allows accepting doctor applications. Requires authorization and probably calls doctorController.acceptdoctor to approve a doctor's application.

/rejectdoctor: Facilitates rejecting doctor applications. Requires authorization and likely calls doctorController.rejectdoctor to decline a doctor's application.

The **appointmentRoutes.js** file handles appointments-related functionalities on the website.

/getallappointments: Retrieves information about all appointments. Requires authorization and likely calls appointmentController.getallappointments to fetch and return appointment data.

/bookappointment: Allows users to book appointments. Requires authorization and probably calls appointmentController.bookappointment to create a new appointment in the database.

/completed: Marks appointments as completed. Requires authorization and presumably calls appointmentController.completed to update the status of the appointment.

These routes collectively enable users to interact with various features on the website, facilitating user management, doctor registration, and appointment scheduling. Overall, the frontend UI interacts with the backend routes through HTTP requests, enabling seamless communication between the user-facing interface and the server-side logic. This integration ensures that users can effectively utilize the website's features while maintaining data integrity and security.

5 Conclusion

To sum up, the Comprehensive BrainGuardian is a major step forward in dealing with the complex challenges facing brain healthcare today. This complete system, which includes predictive modeling, optimized appointment booking, and video consultation facilitation, provides a customized solution that is ready to improve patient outcomes and healthcare efficiency.

Advanced machine learning algorithms can be incorporated into predictive models for brain tumors, which has the potential to transform early detection capabilities. The BrainGuardian system can arm healthcare practitioners with a robust tool to identify potential brain tumors in their early stages. This can significantly enhance treatment effectiveness and positively influence patient outcomes.

Optimizing appointment booking processes for brain health can streamline access to specialized services, reduce waiting times, and enhance the overall patient experience. This aspect of the system aligns with the overarching goal of improving healthcare accessibility and efficiency.

The inclusion of a secured video consultation platform expands the availability of healthcare services by overcoming geographical limitations. This feature not only ensures prompt access to professional medical guidance but also provides an opportunity for a more patient-centered and adaptable healthcare delivery model.

As we look towards the future, the Comprehensive BrainGuardian project is positioned to redefine the standards of neurological care. By blending healthcare expertise with cutting-edge technology, this initiative represents a significant step towards a healthcare landscape that is more integrated, efficient, and focused on patient needs. The impact of this project includes improved patient outcomes, early detection of brain health issues, and a fundamental shift in how neurological conditions are managed. The evolution from problem identification to the proposed solution highlights the transformative power of interdisciplinary collaboration and technological innovation in shaping the future of healthcare.

References

- Accurate brain tumor detection using deep convolutional neural network
- A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images
- Identification of Brain Tumor using Image Processing Techniques