**DOCUMENT**

**Team-3**

**Title:** Iris Tumor Detection Using CNN

**Team Members:**

K. Hariharan

M. Jayasri

C. M. Dharshini

**1. Project Overview**

**What's the Goal?**

The goal of this project is to develop a deep learning-based system for detecting tumors in iris images using Convolutional Neural Networks (CNNs). The system will automate the process of identifying tumors in medical iris scans, improving diagnostic accuracy, and assisting healthcare professionals in early-stage cancer detection.

**What the project will include:**

* Developing and training a CNN model for tumors detection using a labeled dataset of iris images.
* Building a user-friendly interface to upload images and display results.
* Integrating the trained model into a web-based.

**What the project will not include:**

**Medical Diagnosis**:

* The project will not provide a medical diagnosis or predict the health outcomes of detected tumors. It will only identify potential tumor locations based on the iris images, leaving actual diagnosis and treatment decisions to healthcare professionals.

**Why is it Important?**

* **Healthcare Applications**: Early detection of tumors in the iris can aid in diagnosing underlying health conditions such as cancer or other diseases.
* **Automation in Diagnostics**: Automating tumor detection speeds up the process and reduces human error in diagnosis.

**Who's Involved?**

* **Project Leader**: Oversees the project, manages the team, and ensures timely delivery of milestones.
* **Interns**: Develop the user interface and integrate the model into an application.
* **Healthcare Professionals**: Provide domain expertise and guidance on dataset accuracy, medical relevance, and system validation.
* **End Users**: Researchers who will use the system to detect tumors in iris scans.

**2. Requirements Documentation**

**What Does it Need to Do?**

The system should:

* Detect tumors in iris images with high accuracy.
* Display results indicating whether a tumor is present or not, with a confidence score.
* Allow to upload of new iris scan images for tumor detection.
* Provide a user-friendly interface for interaction.

**How Well Does it Need to Work?**

* **Accuracy**: The CNN should achieve at least 90% accuracy on the validation set for tumor detection.
* **Security**: The system should implement secure data handling practices, particularly if sensitive patient data is involved.
* **Scalability**: The model and interface should support future integration with larger datasets or additional diagnostic tools.

**How Will Users Use It?**

**User Stories:**

* **As a medical practitioner**, want to upload iris scan images to detect potential tumors, so they can assess patient’s health quickly.
* **As a researcher**, want to use the system to analyze large sets of iris scans for tumor presence, so they can publish findings on early detection.

**Use Cases:**

1. **Tumor Detection**: The user uploads an iris scan image, and the system processes it, outputting a "tumor detected" or "no tumor detected" result with a confidence score.
2. **Results Visualization**: After processing, the system shows the image with any detected tumor regions highlighted for easier verification by the user.

**Project Plan**

**When Will it Be Done?**

* **Phase 1: Planning (1 weeks)**
  + Define project scope, goals, and roles.
  + Gather medical images and dataset specifications.
* **Phase 2: Development (8 weeks)**
  + Data collection and preprocessing.
  + Model development and training.
  + Design and development of the user interface.
* **Phase 3: Testing**
  + Usability testing with healthcare professionals.
  + Performance optimization.

**What Do We Need?**

* **Personnel:** We need Interns to development with a Experienced Person to lead.
* **Equipment**: Testing hardware such as workstations or laptops for deployment.
* **Software**: Python for CNN model development, Django for the web application and database for storing images.

**How Much Will it Cost?**

* **Development Costs**:
  + Personnel salaries.
  + Cloud computing costs.
* **Testing and Deployment**:
  + Infrastructure and deployment services (e.g., for cloud services, for application deployment).

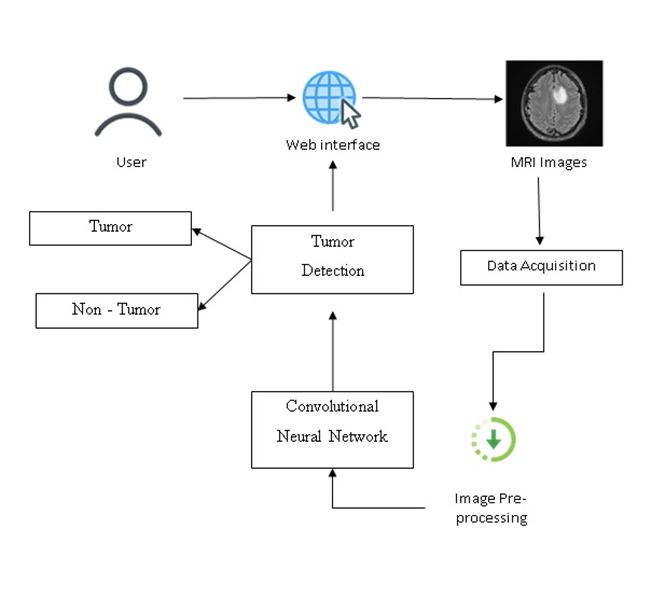
**Architecture and Design Documentation**

**How is it Built?**

The system will use the following architecture:

* **Frontend**: Web or desktop application for user interaction.
* **Backend**: Flask or Django server for handling image uploads and model inference.
* **CNN Model**: Convolutional Neural Network trained on labeled iris tumor datasets.
* **Database**: A relational database (PostgreSQL or MySQL) to store user data, images, and results.

**Diagram**:



**What Does it Look Like?**

* **User Interface**: The user will interact with the web page and upload the image and the result will be displayed.
* **Model Interface**: The CNN model will be integrated into the back-end of the system for processing uploaded images and producing tumor detection results.

**How is Data Stored?**

* **Data Structure**:
  + **Users Table**: Stores basic user info (patients, doctors).
  + **Images Table**: Stores uploaded image metadata (location, type).
  + **Results Table**: Stores predictions.

**5. Testing and Quality Assurance**

**How will we ensure it works?**

To ensure the Iris Tumor Detection system works as expected, we will implement a comprehensive testing strategy across various stages of development. Our testing approach will include the following types of tests:

* **Unit Tests**: Ensure that individual functions, methods, or components work as expected.
* **Integration Tests**: Test how different components of the system (e.g., frontend, backend, model) work together.
* **System Tests**: Validate that the system's functionality meets the specified requirements and user needs.
* **Performance Testing**: Measure how well the system performs under normal and extreme conditions.

**What makes it complete?**

For the project to be considered complete and accepted by stakeholders, the following conditions must be met:

* **Functional Completion**: The system must detect tumors in iris images with a minimum accuracy of 90% on unseen test data.
* **Security and Compliance**: The system must adhere to relevant healthcare data privacy if it involves sensitive patient data.
* **User Feedback**: The system should be user-friendly, with medical professionals reporting that it meets their needs for diagnostic support.
* **Documentation**: Complete system documentation (technical, user manuals, API documentation) should be provided to end-users and developers.

**What if something goes wrong?**

* **User Experience Issues:** Ensure that if something goes wrong (e.g., failed image upload, processing error), the system provides clear, user-friendly error messages.
* **Error Detection and Logging**: Implement automated logging for both backend and frontend to track errors, failed requests, or unexpected behavior.

**6. Deployment and Implementation Plan**

**Where will it live?**

The deployment environment will be a **cloud-based infrastructure** to ensure scalability, flexibility, and ease of maintenance. A **hybrid cloud approach** may be considered, where sensitive data is stored in private cloud environments, and non-sensitive workloads run in public cloud environments.

* **Cloud Service Providers**: AWS, Google Cloud, or Microsoft Azure will be used for hosting the web server and model inference service.
* **Database**: The relational database (PostgreSQL or MySQL) will be hosted on a managed cloud database service.
* **Data Storage**: Iris images and model results will be stored in a secure, scalable cloud storage solution like Google Cloud Storage.

**How will we get it there?**

* Set up the **c**loud infrastructure and containerize the app.
* Develop the model and system backend/frontend.
* Test and deploy to production.
* Ensure security and compliance.
* Monitor, maintain, and update based on feedback.

**What if something goes wrong?**

In the event of a deployment issue:

* **Backup Strategy**: We will maintain backup copies of both the database and model weights in a secure, redundant location to ensure the system can be restored quickly.
* **Rollback Plan**: If a deployment fails or introduces critical issues, the system will be rolled back to the last stable version.
* **Monitoring**: Automated monitoring tools will track system health and alert the team to any issues.

**How will users learn to use it?**

* Provide aninterface with clear instructions.
* Offer an onboarding tutorial or interactive demo.
* Include a help section with FAQs and support options.
* Supply user documentation and video guides.

**7. Maintenance and Support**

**Who's in charge of keeping it running?**

* + **System Administrators**: Responsible for cloud infrastructure, server health, and scaling.
  + **Backend Developers:** Handle server-side functionality.
  + **Frontend Developers**: Manage the user interface and experience.

**What needs to be done to keep it running?**

* **Regular Software Updates**: Update the system periodically to address security vulnerabilities and improve features.
* **Database Maintenance**: Regular checks on database integrity and performance, including backups.
* **Performance Monitoring**: Continuous monitoring of the system’s performance to ensure it meets response time and accuracy benchmarks.

**How will we know what users think?**

* **User Feedback**: Users can provide feedback via the system's interface.
* **Surveys**: Periodic surveys will be sent to medical professionals who are using the system to collect feedback on usability and effectiveness.

**What are our service commitments?**

* **User Support:** Provide timely and responsive customer support for troubleshooting and inquiries.
* **Non-Critical Issues**: Non-critical bugs will be addressed within 7 days.
* **Scalability:** Ensure the system can handle increasing user traffic and data volume without performance degradation**.**

**8. Risk Management**

**What could go wrong?**

* **System Downtime**: The platform may experience outages or unavailability due to server issues.
* **Data Corruption**: Image data or user information could become corrupted or lost during processing or storage.

**How can we prevent problems?**

* **Improve Model Accuracy**: Regularly retrain the CNN model with diverse, high-quality data and continuously evaluate its performance**.**
* **Optimize Performance**: Regularly monitor and optimize system performance, to handle increased demand efficiently.
* **Ensure System Redundancy**: Implement load balancing, failover systems, and regular backups to prevent downtime and data loss.

**What's our backup plan?**

* **Data Backups**: Regular, automated backups of user data and model information, stored securely and offsite (e.g., cloud storage) to prevent data loss.
* **Model Backup**: Store multiple versions of the trained CNN model and related configurations to quickly restore them in case of corruption or errors.

**Who's watching for problems?**

* **Project Manager**: Oversees overall risk management and ensures that mitigation plans are in place.
* **Technical Leads**: Monitor technical risks, such as model accuracy, system performance, and scalability.
* **Security Officer**: Ensures compliance with security standards and oversees data privacy protections.

**9. Security and Privacy**

**Data Protection:**

* **Encryption**: Ensure all sensitive data is encrypted both at rest and in transit using strong encryption methods.
* **Access Control**: Implement role-based access control (RBAC) and multi-factor authentication to limit data access to authorized users only.
* **Data Minimization**: Collect only necessary data and anonymize or pseudonymize it whenever possible to protect user identities.

**Security Measures**

**Regular Security Audits**: Conduct routine security audits and vulnerability assessments to identify and mitigate potential threats.

**Firewalls**: Use firewalls to monitor and protect the system from external attacks or unauthorized access attempts.

**Data Backup and Recovery**: Ensure encrypted data backups are taken regularly and have a disaster recovery plan in place to restore data in case of breach or loss.

**Incident Response Plan**

In the event of a security breach or data compromise:

* **Preparation**: Establish clear roles and responsibilities for the response team and provide training on security threats and how to respond.
* **Containment**: Immediately take action to contain the incident, preventing further damage or data loss.
* **Eradication**: Once contained, remove the cause of the incident and fix any vulnerabilities.
* **Recovery**: Restore affected systems and data from backups, ensuring normal operations resume as soon as possible.

**10. Legal and Compliance**

**Licensing:**

To ensure legal compliance and avoid intellectual property infringement, the following licenses will be required:

* **Software License**: Determine whether the software components (e.g., the website, model, backend) will be open-source or proprietary.
* **Data License:** Ensure the medical data used (e.g., iris images, tumor data) is obtained with proper consent and complies with relevant data privacy regulations.

**Regulatory Compliance**

The system will need to comply with industry-specific regulations, depending on its intended use:

* **FDA Compliance:**

If the system is classified as a medical device (e.g., tumor detection), it must comply with FDA regulations.

* **ISO Standards**:

Specifies quality management systems (QMS) for medical device manufacturers, ensuring safety and effectiveness.

* **Data Privacy Regulations**:

Protects patient health information. The system must ensure data encryption, secure access, and patient consent**.**

**Intellectual Property**

To protect the intellectual property (IP) rights:

* + Copyright for software code, UI/UX design, and algorithms.
  + Patents for novel algorithms, methods, or hardware innovations**.**
  + Licensing third-party libraries and datasets.

**11. Environmental Impact Assessment**

**Sustainability**

The project will aim to minimize its environmental impact by:

* Environmental sustainability is achieved through energy-efficient algorithms and green data centers.
* Economic sustainability is supported by cost-effective solutions, scalability, and the use of open-source tools.

**Green IT Practices**

The system will employ the following **Green IT** practices:

* + Cloud computing and server virtualization to optimize resource usage.
  + Carbon footprint monitoring to track and reduce emissions.
  + Green data centers that use renewable energy and energy-efficient cooling.

**12. User Documentation**

**User Manuals**

* **System Overview**: Introduction to the Iris Tumor Detection system and its capabilities.
* **Step-by-Step Instructions**: Detailed guidance on how to upload iris images, interpret results.
* **Security and Privacy**: Instructions on how users can maintain the security of their data (e.g., login procedures, role-based access).

**Online Help and Tutorials**

The system will include several online resources to assist users, including:

* **FAQs**: A set of frequently asked questions addressing common queries.
* **Video Tutorials**: Step-by-step video guides on using the system, including how to interpret results and best practices for iris scan imaging.

**User Interface Design**

The **user interface (UI)** will be designed with the following goals:

* **Login Screen**: Simple authentication with role-based access.
* **Image Upload Page**: Easy-to-use drag-and-drop area for uploading iris images.
* **Tumor Detection Results Page**: Displays detected tumor areas on iris images.

**13. Project Management and Monitoring**

**Project Planning**

The project will involve the following steps:

* **Clear phases** Initiation, Planning, Execution, Monitoring, Closing.
* **Well-defined roles and resources**, including team members and technologies.
* **Budget allocation** for personnel, hardware, software, and miscellaneous costs.
* **Risk management** strategies to mitigate data, model, technical, and timeline challenges.

**Resource Allocation:**

* Human resources (project managers, developers, data scientists, designers, QA, healthcare experts).
* Hardware resources (development workstations, cloud storage).
* Software resources (backend and frontend technologies, databases).

**Risk Management**

* Assessing risks by evaluating their likelihood and impact.
* Monitoring and controlling risks with regular reviews and tracking tools.
* Preparing contingency plans to handle potential project disruptions effectively.

**14. Testing and Quality Assurance**

* **Unit Testing**: Testing individual components (e.g., functions, methods) of the code to ensure they work correctly in isolation.
* **Integration Testing**: Verifying that different system modules (e.g., model, backend, UI) interact seamlessly and that data flows correctly between them.

**System Testing**: Testing the entire system end-to-end to ensure all parts work together and meet the overall requirements.

**Quality Assurance (QA)**

* **Accuracy**: Ensuring that the Iris Tumor Detection System correctly identifies tumors in iris images with high precision and minimal false positives or negatives.
* **Reliability**: The system must be stable and function without unexpected crashes or issues, ensuring users can trust it for diagnosing and detecting tumors.
* **Usability**: The user interface (UI) should be intuitive, user-friendly, and designed to meet the needs of healthcare professionals.