IRIS TUMOR DETECTION USING CNN

PROJECT DOCUMENTATION

Team no. 6
Team Members –
Keyuri Patil
Jyothika Kakulapati

1. Project Overview

Goal

The goal of the "Eye Iris Tumour Detection" project is to develop a CNN-based system capable of detecting and classifying iris tumours in medical images, supporting healthcare professionals in early diagnosis and intervention. The project will focus on building a high-accuracy classification model and a user-friendly interface.

The project will include:

- **Image Processing and Classification**: The system will use a CNN model to analyse uploaded iris images and detect possible tumour indicators.
- **User-Friendly Interface**: A web-based interface will be developed, allowing users to interact with the system easily.
- **Secure Data Management**: Data protection and encryption will be employed to ensure secure handling of medical images and user information.
-

The project will not include:

- **Real- time Monitoring**: The system is designed for diagnostic purposes only and does not support continuous monitoring of the eye.
- **Direct Clinical Integration** with Medical Equipment: The project is currently limited to so ware development and image analysis. Integra ng this tool with existing clinical hardware (such as digital fundus cameras) for automated capture and analysis would be a potential future extension.
-

Importance

- Early Detection of Tumours: By identifying tumours at an early stage, the project aims to contribute to improved treatment outcomes.
- Enhancing Diagnostic Accuracy: Using deep learning for tumour classification improves diagnostic precision, reducing human error.
- Accessibility: Provides users with a quick, accessible tool for understanding iris health.

Key Stakeholders

- Project Interns: Oversees development, resource allocation, and project timelines.
- Software Developers: Build the frontend and backend systems for the detection platform.
- End-Users: Users who will use the platform for diagnosis.

2. Requirements Documentation

Core Features and Functions

- **Image Upload and Analysis**: Users can upload high-resolution images of the iris, which the system processes to detect the presence of tumours.
- **Prediction Results**: Provides a result, whether the tumour detected or not.

Quality Standards

- **Performance:** The model must achieve a high degree of accuracy and precision (targe ng >90% accuracy).
- **Security**: Users data images are encrypted and stored securely.
- **Usability:** Simple, intuitive UI for easy navigation for users.

Business Objectives

- **Support Early Diagnosis**: Helps healthcare providers in early and accurate diagnosis, aligning with company goals of innovation in medical AI.
- **Increase Accessibility**: Democratizes access to medical-grade diagnostic tools.
- **Promote Technology Integration in Healthcare**: Fosters the adoption of AI in healthcare, aligning with broader business goals of advancing AI-driven solutions.

User Interaction

- The main user interaction involves uploading an iris image to detect a tumour. Here's a brief use case:
- **Login:** The user logs into the system securely.
- **Image Upload**: They navigate to the "Tumour Detection" sec on, select an iris image from their device, and upload it.
- **Prediction Result**: The system's CNN model analyses the image and displays a result, showing whether a tumour is detected or not.
- **Logout**: The user logs out a er reviewing the results.
- If any issue occurs (e.g., incorrect image format), the system prompts the user to resolve it, ensuring smooth opera on.

3. Project Plan

1. When Will It Be Done?

The project will be divided into several phases with estimated timeframes for each:

- Planning (2 weeks):
    - Define project requirements, objectives, and success criteria.
- Development (5 weeks):
    - Week 1-2: Set up the backend for photo upload functionality, user registration/login, and storing images.
    - Week 3-4: Develop the CNN model, including training on a dataset of iris images (both normal and with tumours).

- Week 5: Integrate the model with the backend to process uploaded images and display results.

- Testing (2 weeks).
- Deployment (1 week):
  - Deploy the web application to a cloud server or local hosting environment. o Set up the live database and ensure everything is connected and working.
- Post-Deployment (1-2 weeks):
  - Monitor performance, fix any bugs, and refine user interface based on feedback. Total Timeline: 10-12 weeks

2. What Do We Need?

The resources required for the project include:

- Equipment:
  - Computers for development (laptops/desktops).
- Software:
  - Python (for building and training the CNN model).
  - Flask/Django (for web development framework).
  - TensorFlow/Keras/PyTorch (for implementing and training the CNN).
  - HTML, CSS, JavaScript (for frontend development).
  - Database (MySQL/PostgreSQL for user and image data storage).

4. Architecture and Design Documentation

How is it Built?

The system follows a client-server architecture, where the user interacts with the client-side interface (web-based) to upload an iris image, and the server-side handles the processing of the image and the tumour detection using a trained CNN model. Here is an overview of the system components:

- **Client-Side (Frontend):**
  - Web Interface: Allows the user to register, log in, and upload iris images. It interacts with the server-side through HTTP requests (using RESTful APIs or AJAX).
  - User Interface: Displays the result of the tumor detection (whether a tumor is present or not) a er processing the uploaded image.
- **Server-Side (Backend):**
  - Web Server: A Flask or Django server running the application logic. It handles requests from the client-side and interfaces with the machine learning model.
  - Model Inference: A Convolutional Neural Network (CNN) model is hosted on the server. When an image is uploaded, it is sent to the server, which processes it using the CNN model to predict whether a tumour is detected in the iris.
  - Data Handling: The backend handles user data (authentication, image metadata) and the result of tumour detection.
- **Machine Learning Model:**
  - CNN Model: A deep learning model (built using TensorFlow/Keras/PyTorch) is trained to detect tumours in iris images. The model receives an uploaded image, processes it, and returns a prediction of whether a tumour is present.

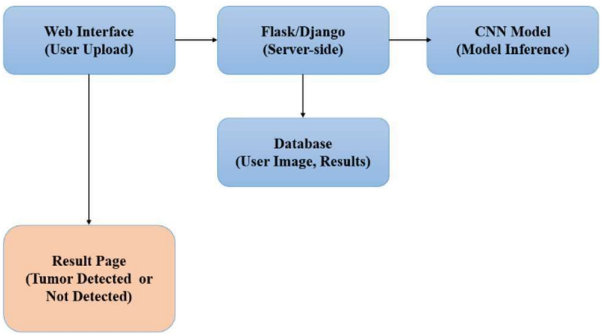
- **Database:**
    - User Data: Stores user information (such as usernames, email, passwords).
    - Image Metadata: Stores information about uploaded images (e.g., file names, user IDs) and the results of the tumour detection (whether a tumour was detected or not).

What Does It Look Like?
Here is a breakdown of the system's architecture with accompanying diagrams:
1. **Web Interface (Client-Side):**
    - Home Page: A landing page with options to log in or register.
    - Upload Page: A page where the user uploads their iris image.
    - Result Page: Displays the result of tumour detection (e.g., "No Tumour Detected" or "Tumour Detected").
2. **Backend (Server-Side):**
    - API Endpoint: A REST API endpoint that receives the image and triggers the CNN model inference.
    - Image Processing: Preprocess the image (resize, normalize) before passing it to the model.
    - Model Inference: The CNN model analyses the image and returns the result to the frontend.

Model Architecture:
- CNN Model: A convolutional neural network architecture that processes the iris image and classifies it as either "Tumour Detected" or "No Tumour Detected."

Database:
- User Table: Stores details like username, email, and password.
- Image Table: Stores uploaded image information (file name, user ID, result).
- Result Table: Stores the outcome of the tumour detection (tumour detected or not).

Flow Diagram:



5. Testing and Quality Assurance

Ensuring Functionality
Our testing approach includes unit, integration, system, and acceptance testing to validate every component:
- **Unit Testing**: Verifies individual functions like image upload and user authentication
- **Integration Testing**: Ensures that frontend and backend components (e.g., file upload to model prediction) work together using Postman and Selenium.

- **System Testing**: Tests the complete workflow, covering performance, security, and end-user experience with tools like JMeter and Selenium.
- **Acceptance Testing**: Validates real-world functionality to ensure the system meets stakeholder expectations, focusing on accuracy and usability.

Key Test Cases

- **User Authentication**: Validates registration, login, and error handling for incorrect inputs.
- **Image Upload**: Confirms proper image upload, format validation, and error handling.
- **Model Prediction**: Tests model's accuracy in detecting tumors using known data.
- **Database Operations**: Verifies integrity in user data, image metadata, and result storage.
- **Performance Testing**: Ensures the system can handle multiple users and high loads.
- **Security**: Checks for vulnerabilities like SQL injection and file upload validation.

Completion Requirements

For the project to be considered complete, it must meet these conditions:

- **Model Accuracy**: CNN model achieves a minimum 90% detection accuracy.
- **Usability and Reliability**: Users can seamlessly upload images and get results without errors.
- **Security**: User data is secure, and potential vulnerabilities are addressed.
- **Documentation**: Comprehensive documentation for setup and usage is provided.
- **Acceptance Testing**: Passes stakeholder review based on functional and performance benchmarks.

Testing Schedule

- **Unit Testing**: Ongoing during development.
- **Integration Testing**: Starts after frontend-backend integration (Week 4-5).
- **System Testing**: Scheduled for Week 7-8, covering end-to-end functionality.
- **Acceptance Testing**: Final testing with stakeholders before deployment (Week 9-10).

Continuous Integration (CI): CI tools like GitHub Actions will automate unit and integration tests on every code commit.

Issue Tracking and Resolution

Tracking and Fixing: Issues are addressed and tested with regression tests to ensure fixes don't introduce new problems.

6. Deployment and Implementation Plan

Objective: Launch the Iris Tumor Detection system in a way that ensures reliability, scalability, and ease of access.

Deployment Environment

- Primary Environment:
  The system will be hosted on a cloud platform like AWS or Azure. This setup offers flexibility for scaling and is easily accessible, which suits a growing user base.
- Growth Considerations:

As more users access the system and data storage needs increase, we'll plan for extra storage space and processing power to handle the additional load.

Deployment Strategy

- Launch:

We'll start with a pilot release to a small, select group of users. This approach allows us to test the system in a real-world setting while minimizing risk.

- Rollout:

Following the pilot, we'll gradually expand access in phases, monitoring each phase to ensure all components are functioning as expected.

- Continuous Delivery:

A CI/CD (Continuous Integration/Continuous Delivery) pipeline will be set up. This pipeline allows us to push updates regularly and automatically, keeping the system up-to-date without major downtime.

Backup and Rollback Strategies

- Backups:

We'll perform regular backups of all essential data, including the database and application files. This ensures data recovery in case of any failure.

- Rollback:

We'll have an automated rollback process. If an update causes issues, we can quickly revert the system to its previous stable version to minimize disruption.

User Training and Onboarding

- Documentation:

We'll provide user manuals and quick-start guides that explain how to use the system step-by-step, making it easy for new users to get started.

- Training Sessions:

Webinars or training videos will be offered to walk users through the system's features, helping them understand how to use it effectively.

7. Maintenance and Support

Who's in Charge of Keeping it Running?

- A small support team, including developers and IT staff, will be responsible for keeping the project running smoothly. They'll handle updates, fix issues, and help users if needed.

What Needs to be Done to Keep it Running?

- Regular Updates: Make sure the software stays up-to-date and compatible with other systems.
- Bug Fixes: Quickly fix any issues that may come up, like broken links, errors on the pages, or model glitches.
- Performance Monitoring: Regularly check the system's speed and responsiveness to ensure it works well for users.
- Security Checks: Keep user data safe by performing regular security checks and updates.
- How Will We Know What Users Think?
- Collecting Feedback: Users can give feedback through a form on the website or email, which the support team will review.
- Using Feedback: Feedback will be reviewed every few months, and useful suggestions will be used to improve the system in future updates.

What Are Our Service Commitments?

- System Uptime: We aim to keep the system running 99% of the time.
- Response and Resolution Times: For critical issues:
- Response Time: We'll respond to major issues within 2 hours.
- Resolution Time: Important issues will be fixed within a day, and minor fixes will be addressed in the next scheduled update.

8. Risk Management

What Could Go Wrong?

1.Technical Issues: Problems with the image detection model or login pages.
2.Operational Issues: Possible website downtime due to server issues.
3.Security Risks: Unauthorized access to user data.
4.Budget Constraints: Not enough funding for regular updates or improvements.

How Can We Prevent Problems?

- For Technical Issues: Run regular tests on the model and website to catch problems early.
- For Operational Issues: Have a backup server ready to minimize downtime if there are any server issues.
- For Security Risks: Use data encryption, secure logins, and follow best practices for data protection.
- For Budget Constraints: Set aside extra funds and prioritize essential features and fixes.

What's Our Backup Plan?

- Plan for Major Issues: If there's a major problem, the support team will restore the system using backups. The system can quickly switch to a backup server to keep running if needed.
- Who's Watching for Problems?
- Risk Monitoring Team: A specific person or small team will be in charge of regularly checking for issues, tracking risks, and making sure everything is running as expected. They'll coordinate with developers to handle any problems that arise.

9. Security and Privacy

Data Protection

Encryption: Protect sensitive data, like patient health information and images, by using strong encryption (such as AES-256). This keeps data secure whether it's stored or being transmitted.

Access Controls: Use role-based access controls (RBAC) to limit who can view sensitive data. Only authorized medical staff and administrators should access diagnostic results.

Data Privacy Regulations: Follow privacy laws like GDPR (for Europe) and HIPAA (for the US) when handling patient data. This includes collecting only necessary data, anonymizing it when possible, and getting clear patient consent.

Security Measures

Firewalls: Set up firewalls to protect servers and databases from unauthorized access and cyber attacks.

Intrusion Detection Systems (IDS): Install IDS to monitor the system and alert you if any unusual or suspicious activity is detected.

Regular Security Audits: Schedule regular security audits and scans to check for weaknesses and keep the system safe.

Incident Response Plan

Preparation: Define who will handle different tasks during a security incident, such as communication and investigation.

Detection and Analysis: Set up monitoring tools to quickly detect any unauthorized access or data breaches.

Containment, Eradication, and Recovery: Have a clear plan to contain a breach, remove any threats, and restore affected data or systems.

Post-Incident Review: After an incident, review what happened to improve security measures and update protocols.

10. Legal and Compliance

Licensing

Software and Hardware Licenses: Make sure that all software libraries, AI tools, and hardware used in the iris tumor detection system are legally licensed. For open-source software, verify that licenses are compatible with your project, especially if it might be used commercially.

Regulatory Compliance

Medical Device Regulations: If the project is used in clinical settings, follow regulations like FDA standards (in the US) or ISO standards (such as ISO 13485 for medical devices), especially if it could impact medical decisions.

Data Protection Compliance: Regularly check that the system meets data protection rules like GDPR and HIPAA. This includes keeping clear documentation on how data is handled and running regular audits to ensure compliance.

Intellectual Property

1. **Patents**: If your iris tumor detection algorithm is innovative, consider patenting it to protect your intellectual property.
2. **Trademarks**: If the project will have a brand name, logo, or unique identity, trademark these elements to protect the brand.
3. **Copyrights**: Document your code, documentation, and other assets to claim copyright over your original work.

11. Environmental Impact Assessment

Sustainability: To minimize environmental impact, we'll leverage cloud computing for energy-efficient infrastructure. Efficient coding and optimized model processing will reduce computational load.

Green IT Practices:

- Energy-Efficient Hardware: Cloud services with efficient, renewable-powered data centers will be prioritized.
- Virtualized Infrastructure: Virtual servers will reduce the need for physical hardware.
- Scalable Cloud Services: Auto-scaling will adjust resource use based on demand to prevent energy waste.
- Data Retention Policies: Old data will be deleted automatically to reduce storage needs.

12. User Documentation

User Manuals:

- User Guide: Covers all user actions from registration to result interpretation with screenshots and troubleshooting.
- Technical Manual: For administrators, covering system setup and maintenance.
- Maintenance Guide: Instructions for model updates, data management, and performance optimization.

Online Help and Tutorials:

- FAQs and Step-by-Step Tutorials provide answers and walkthroughs for common tasks.
- Video Guides demonstrate essential functions visually.
- Contact Support for unresolved issues.

User Interface Design:

- Simplified Layout: A clean design with clear buttons guides users.
- Real-Time Feedback: Users see upload progress and receive alerts for errors.
- Accessible Design: Keyboard navigation and readable fonts ensure usability for all.
- This approach ensures an eco-friendly, easy-to-use, and well-supported application for detecting iris tumours.

13. Project Management and Monitoring

Project Planning:

Develop a comprehensive project plan that includes tasks such as data collection, model training, and interface design, along with detailed timelines and resource allocations. Assign team members to tasks based on expertise, ensuring efficient use of resources for each phase.

Risk Management:

Potential Risks:

- **Model Accuracy Issues:** The model may fail to achieve desired accuracy.
- **System Integration Errors**: Frontend and backend may not sync correctly.
- **Security Vulnerabilities**: Risks associated with image uploads.

Mitigation Strategies:

- Use cross-validation and data augmentation to improve model accuracy.
- Conduct thorough integration testing early in the process.
- Plan cloud configurations in advance and test deployment in a staging environment.
- Implement security measures like file type validation and HTTPS.

14. Testing and Quality Assurance

Unit Testing

Focus: Test individual components like:

- **Image Upload**: Ensure correct formats (e.g., PNG, JPEG) are accepted and files are uploaded successfully.
- **Model Prediction**: Validate that the CNN model accurately detects tumours on test images.

Integration Testing

- **Focus:** Test the interaction between:
- **Frontend and Backend**: Validate smooth data flow from user uploads to model predictions and result display.
- **Backend**: Ensure uploaded images and prediction results are correctly logged .

This structured project management and testing approach ensures timely delivery and a reliable, user-friendly system for iris tumor detection.