Deep Learning for Cardiovascular Risk Detection from Retinal Image

Smt. Kavya SN[1] , MS. Bhavani NB[2] , MS. Bhuvana S[3] , MS. Deekshitha V[3] , MS. Sushma K U[4]
[1]Assisstent Professor, Department Of Computer Science, MMK and SDM MMV, MYSURU
[2,3,3] Student, Department Of Computer Science, MMK and SDM MMV, MYSURU

Abstract- **Cardiovascular diseases (CVDs) remain a leading cause of global morbidity and mortality. Early detection and intervention are crucial for improving patient outcomes and reducing the burden on healthcare systems. Recent research suggests a potential link between retinal vascular changes and cardiovascular health. Retinal images offer a non-invasive means to assess microvascular abnormalities, making them an attractive source of data for predictive modeling. This project focuses on developing a machine learning model, specifically using Recurrent Neural Networks (RNNs), to analyze retinal images and detect patterns indicative of heart diseases. RNNs are well-suited for processing sequential data, making them suitable for capturing temporal dependencies in the retinal images and improving the predictive accuracy of the model.**

I. INTRODUCTION

Heart disease, also known as cardiovascular disease, refers to a class of diseases that involve the heart or blood vessels. It is a broad term encompassing various conditions that affect the cardiovascular system, including coronary artery disease, heart failure, arrhythmias, and valvular heart disease. The most common type of heart disease is coronary artery disease, which occurs when the coronary arteries become narrowed or blocked due to atherosclerosis, leading to reduced blood flow to the heart muscle. This can lead to heart attacks, which can be life-threatening if not treated promptly. The utilization of retinal images as a diagnostic tool for heart disease has gained attention in recent years due to the potential correlation between retinal characteristics and cardiovascular health. The retina, as a neural tissue, shares vascular similarities with the cardiovascular system, and abnormalities in retinal vessels may indicate underlying heart conditions. Studies have shown that changes in retinal blood vessels, such as narrowing or irregularities, can be indicative of hypertension, atherosclerosis, and other cardiovascular conditions.

Retinal imaging offers a non-invasive and cost-effective method for detecting cardiovascular disease. By analyzing retinal images using machine learning algorithms, such as Recurrent Neural Networks (RNNs), it may be possible to identify patterns and abnormalities in retinal vessels that are indicative of heart disease. This could enable early detection and treatment of cardiovascular conditions, potentially reducing the risk of complications and improving patient outcomes.

The aim of this machine learning project is to leverage RNNs for the detection of heart diseases through the analysis of retinal images. By developing an accurate and efficient system for analyzing retinal images, we hope to contribute to the early diagnosis and timely intervention of heart disease, ultimately improving cardiovascular health outcomes.

II. SCOPE

Healthcare professionals could use the developed model as an auxiliary tool for early detection and screening of heart diseases during routine eye examinations. The project could be integrated into telemedicine platforms, allowing for remote monitoring of patients' cardiovascular health through retinal images captured using portable devices or smartphones.

General practitioners and primary care physicians could benefit from the model as a supplementary diagnostic tool.The project could be utilized in public health campaigns aimed at increasing awareness and promoting early detection of heart diseases.

III. OBJECTIVES

The primary objectives of this machine learning project are to collect and preprocess a diverse dataset of retinal images annotated with cardiovascular health status, and to design and implement a

Recurrent Neural Network (RNN)-based model for the analysis of sequential retinal data. The goal is to optimize the model for sensitivity and specificity in heart disease detection, and to evaluate its performance using relevant metrics.

To achieve this, the project will involve gathering a comprehensive dataset of retinal images from various sources, including hospitals, clinics, and research studies. These images will be annotated with cardiovascular health status, including the presence or absence of heart disease, as well as the severity of the disease. The images will then be preprocessed to ensure quality, consistency, and suitability for analysis.

The RNN-based model will be designed to analyze the sequential retinal data and identify patterns and abnormalities that are indicative of heart disease. The model will be trained on the collected dataset and optimized to achieve high sensitivity and specificity in heart disease detection. The performance of the model will be evaluated using relevant metrics, including accuracy, precision, recall, and F1-score.

The expected outcome of this project is to contribute to the early detection of heart attack risk through the analysis of retinal images. By enabling early detection and treatment, the project aims to improve patient outcomes and reduce the risk of complications. Additionally, the project aims to provide clinicians with a valuable tool for making informed decisions about patient care and treatment. Overall, the project has the potential to make a significant contribution to the field of cardiovascular healthcare, enabling early detection and treatment of heart disease and improving patient outcomes.

IV. PROPOSED METHODOLOGY

4.1 Data Collection:
Collect a diverse and representative dataset of retinal images from individuals with varying cardiovascular health statuses. Ensure the dataset represents different demographics, ages, and risk factors. Annotations indicating whether each individual has a known cardiovascular condition or is healthy should be included.

4.2 Data Preprocessing:
Preprocess the data to ensure uniformity and remove irrelevant information. Ensure uniformity in the dataset by standardizing the resolution, color channels, or any other relevant parameters of the retinal images. Remove irrelevant information or artifacts that may not contribute to the heart disease detection task. This step may involve noise reduction, image cropping, or masking.

4.3 Model Development:
Design and implement an RNN-based architecture suitable for processing sequential retinal image data. Train the RNN-based model on the prepared dataset. During training, the model learns to recognize patterns and relationships within the sequential retinal images and their corresponding labels. Optimize the model for sensitivity and specificity. Sensitivity ensures the model can detect true positives, while specificity minimizes false positives.

4.4 Model Evaluation and Validation:
Assess the performance of the developed model using relevant metrics. Validate the model on independent datasets to ensure generalizability. Evaluate the model using metrics such as accuracy.

4.5 Testing and deployment:
Use the testing dataset to evaluate the final performance of the trained model on unseen data. Once satisfied with the model's performance, it can be deployed to make predictions on new retinal images and detect heart disease.The proposed system will be developed using Python and Flask, a web application framework. Flask is a lightweight and modular web application framework for Python. It is known for its simplicity, making it a popular choice for developing small to medium-sized web applications. Flask provides the necessary tools for routing, templating, and handling HTTP requests.
By following this proposed methodology, the aim is to develop a robust RNN-based model for heart disease detection using retinal images, ensuring data quality and model performance are prioritized throughout the process.

V. ALGORITHM – RNN

In the context of detecting heart diseases using retinal images, Recurrent Neural Networks (RNNs) are pivotal for analyzing sequential data, such as the vascular patterns present in the retina. RNNs, equipped with memory mechanisms, can capture dependencies over time, enabling them to recognize subtle changes in retinal features indicative of cardiovascular health.

By leveraging Long Short-Term Memory (LSTM) cells to address long-term dependencies, RNNs extract meaningful features from retinal sequences, facilitating the identification of abnormalities associated with heart diseases. Through iterative training and optimization, the RNN learns to map retinal images to disease labels, contributing to the integration of diagnostic systems for early detection and intervention. In essence, RNNs play a critical role in enhancing the accuracy and efficiency of heart disease detection from retinal images, paving the way for proactive management of cardiovascular health.

VI. DATASET

The dataset used in this project consists of 3662 retinal images. This dataset is taken from Kaggle to train the model. The dataset categorized into different risk levels associated with diabetic retinopathy (DR) and other cardiovascular conditions. Diabetic retinopathy is a common complication of diabetes and is characterized by damage to the blood vessels in the retina. The dataset includes images representing various stages of DR, ranging from mild to severe, as well as images indicating the absence of DR.

Additionally, the dataset may contain images representing proliferative diabetic retinopathy (PDR), the most advanced stage of DR characterized by the growth of abnormal blood vessels in the retina. Each retinal image in the dataset is annotated with its corresponding risk category, allowing for supervised learning algorithms to be trained for heart disease detection. This diverse dataset enables the development and evaluation of machine learning models aimed at leveraging retinal images for early detection and classification of heart diseases, providing valuable insights into the relationship between retinal characteristics and cardiovascular health.

VII. MODULES

This web application consists of following modules

Register:

This module allows new users to sign up by entering their personal information such as name, email, and password. The data is typically stored in database after validation. It ensures each user gets a unique identity in the system and is a necessary step before accessing the systems features.

Login:

The login module authenticates users using their credentials such as username, email and password. Upon successful login, users gain access to functionalities based on their roles (e.g., admin or regular user).

Upload Dataset:

This module is accessible to the admin and is used to upload retinal image datasets which are required for training the Recurrent Neural Network (RNN). These images form the core data used in building the heart attack detection model.

Train Module:

This part of the system is responsible for initiating the training process of the RNN model using the uploaded retinal images. It involves data preprocessing, feeding the data into the RNN, and tuning model parameters.

Save Model:

Once the model is trained, this module allows saving the trained weights and architecture to disk or cloud storage for future predictions without retraining.

View Performance:

This module presents evaluation metrics to assess the effectiveness of the trained model. It includes performance indicators such as accuracy, precision, recall, F1-score, and loss over epochs.

Input Test Data:

This module allows users to input new retinal images for the purpose of analysis and risk detection. It applies the trained model to unseen data.

View Detection:

Once test data is processed, this module displays the model's output—i.e., whether the user is at risk of a heart attack based on retinal image analysis.

VIII. SOFTWARE TESTING

Introduction

Software testing is a process of executing a program or application with the intent of finding the software bugs. Software testing is a critical element of software quality assurance and represents the ultimate process to ensure the correctness of the product. The quality product always enhances the customer confidence in using the product thereby increasing the business economics. In other words, a good quality product means zero defects, which is derived from a better quality process in testing.

Testing the product means adding value to it by raising the quality or reliability of the product. Raising the reliability of the product means finding and removing errors. Hence one should not test a product to show that it works; rather, one should start with the assumption that the program contains errors and then test the program to find as many of the errors as possible.

The main objective of testing is to find defects in requirements, design, documentation, and code as early as possible. The test process should be such that the software product that will be delivered to the customer is defect less. All Tests should be traceable to customer requirements. Test cases must be written for invalid and unexpected, as well as for valid and expected input conditions.

A necessary part of a test case is a definition of the expected output or result. A good test case is one that has high probability of detecting an as-yet undiscovered error.

Types of testing done on the system

Manual Testing

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

Different stages for manual testing:

Unit Testing

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

Tests that are performed during the unit testing in app are explained below:

1) Module Interface test: In module interface test, it is checked whether the information is properly flowing in to the program unit (or module) and properly happen out of it or not.

2) Boundary conditions: It is observed that much software often fails at boundary related conditions. That’s why boundary related conditions are always tested to make safe that the program is properly working at its boundary condition’s.

E.g. In case of if...else if... else... construct all the conditions are checked in the app. In case of loops, it is checked to see that the loops are not infinite and terminate once the condition becomes false.

3) Error handling paths: These are tested to review if errors are handled properly.

Integration Testing

Integration testing is a crucial phase of software testing where individual units or modules are combined and tested as a group to ensure they work together seamlessly. This type of testing helps identify issues that may arise when different modules interact with each other.

Types of Integration Testing

1. Bottom-up Integration Testing: This approach starts with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. It's like building a pyramid from the base up.

2. Top-down Integration Testing: This approach starts with the overall system, and then individual units are tested. It's like starting with the complete picture and then drilling down to the details.

In the Bottom-up approach:

1. Unit Testing: Each unit or module is tested individually to ensure it works correctly.

2. Module Integration: Units are combined into modules or builds, and testing is performed on these integrated modules.

3. Progressive Testing: Testing progresses from lower-level modules to higher-level modules.

Activities in Integration Testing

When all the different modules were integrated in the app, the following activities were performed:

1. Transition from one screen to another: Testing the flow of the application, ensuring that the transition between screens is smooth and works as expected.

2. Interaction between modules: Testing how different modules interact with each other, ensuring that data is exchanged correctly and functionality works as expected.
3. Error handling: Testing how the application handles errors and exceptions when different modules interact with each other.

Benefits of Integration Testing
1. Early Detection of Defects: Integration testing helps detect defects early in the development cycle, reducing the overall cost of fixing defects.
2. Improved Quality: Integration testing ensures that the different modules work together seamlessly, improving the overall quality of the application.
3. Reduced Risk: Integration testing reduces the risk of defects and issues that may arise when different modules interact with each other.

System Testing
System testing is a critical phase of software testing where the entire system is tested as a whole, ensuring it meets the specified Quality Standards. This type of testing evaluates the system's functionality, performance, and reliability.

Objectives of System Testing
1. Verify System Functionality: Ensure the system works as expected and meets requirements.
2. Identify Defects: Detect defects and issues that may have been missed during unit and integration testing.
3. Ensure Quality Standards: Verify the system meets specified Quality Standards.

Activities in System Testing
1. Testing for All Possible Inputs: Testing the system with various inputs, including valid and invalid data.
2. Executing Test Cases: Running different test cases to ensure the system behaves as expected.
3. Checking for Crashes and Unexpected Behaviour: Verifying the system doesn't crash or exhibit unexpected behaviour.

Types of System Testing
1. Functional Testing: Testing the system's functionality.
2. Performance Testing: Testing the system's performance under various conditions.
3. Security Testing: Testing the system's security features.

Benefits of System Testing
1. Ensures System Quality: System testing ensures the system meets Quality Standards.
2. Detects Defects: System testing detects defects and issues.
3. Improves User Experience: System testing ensures the system is user-friendly.

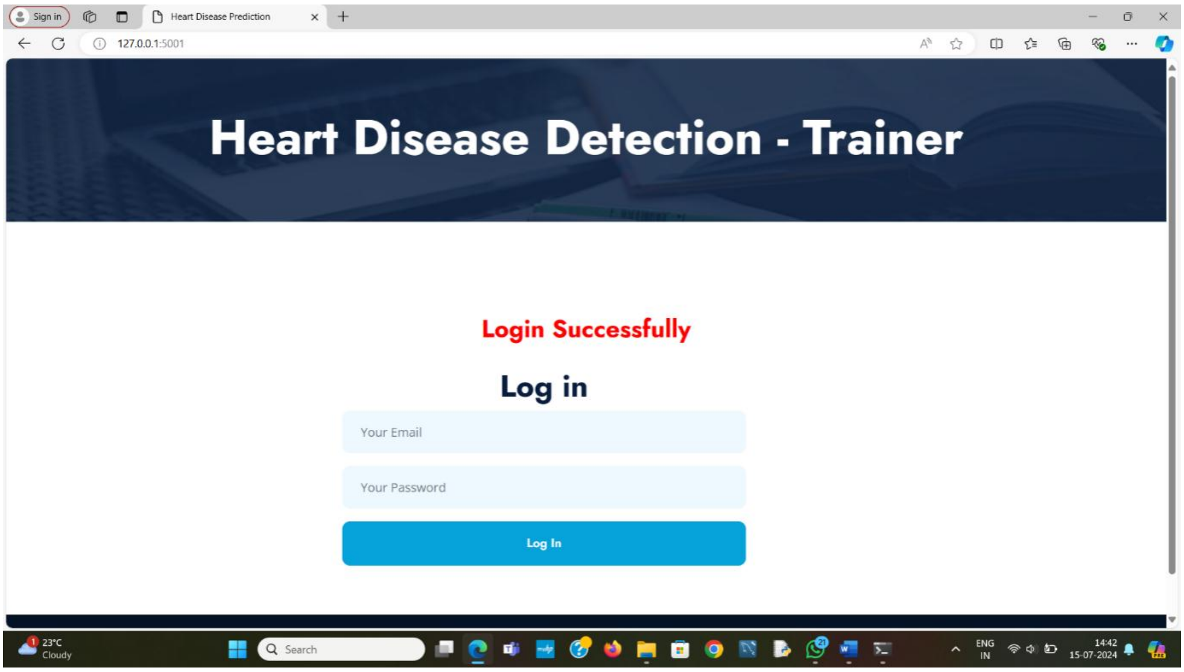
System Testing Deliverables
1. Test Plan: A document outlining the testing approach.
2. Test Cases: Specific scenarios tested.
3. Test Results: Results of testing.

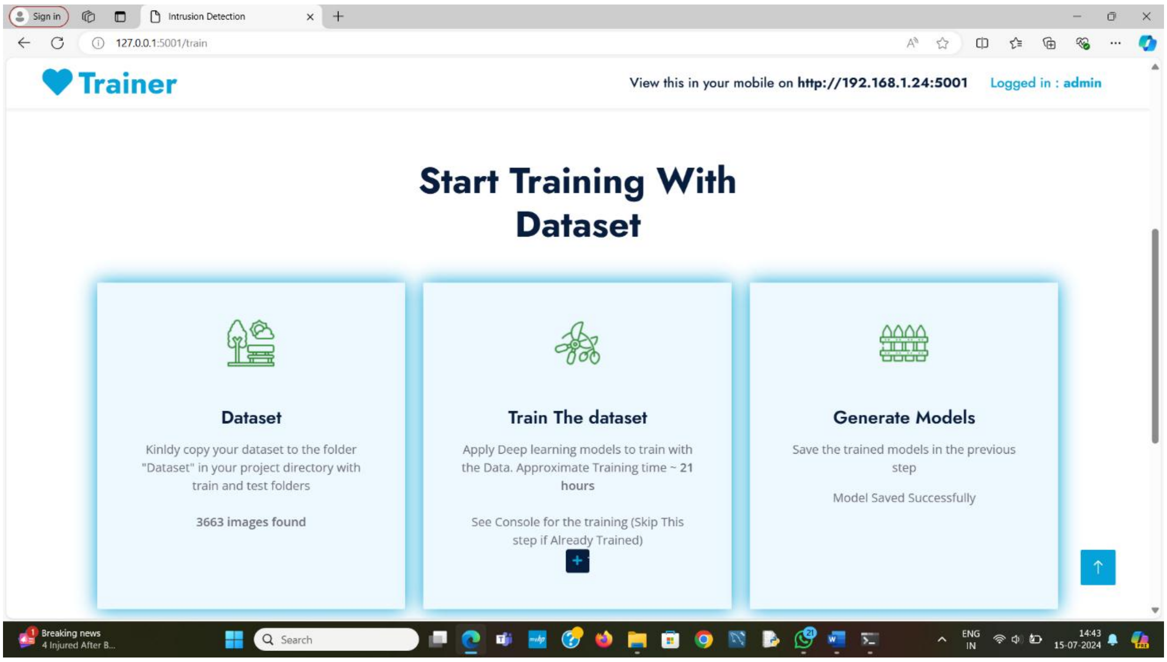
Test Cases

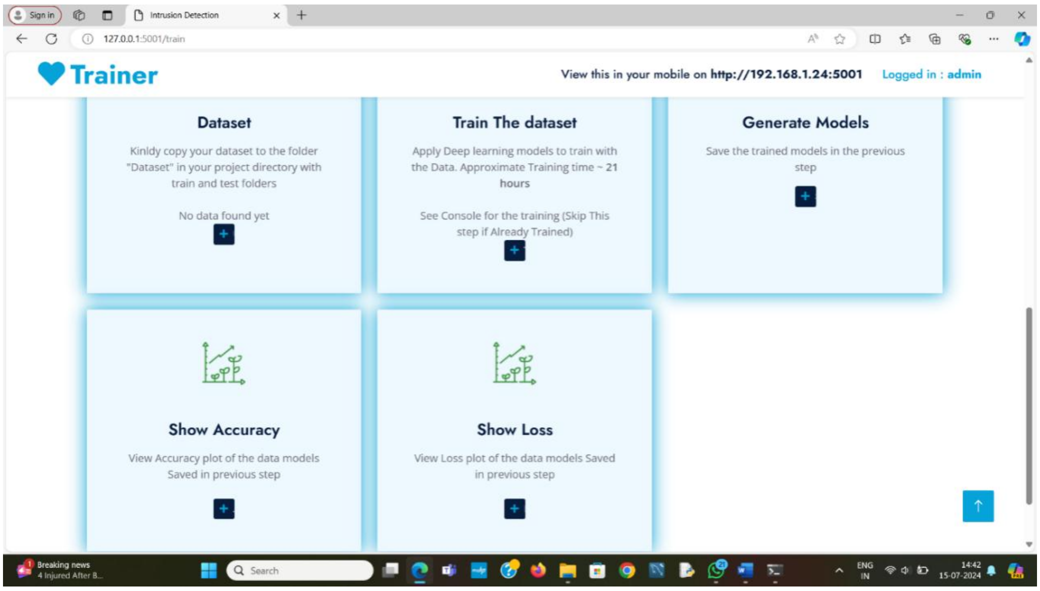
TC 01	Data Collection	To import data from location	Import from file location	Valid location and file name	import data successful	Import data successful	Pass
	Data Collection	To import data from location	Import from file location	Invalid location and file name	Import data unsuccessful	Import data unsuccessful	Pass
TC 02	Data pre-processing	To verify that data has pre-processed	Enter the data file imported	Valid detail	pre-processing successfully	pre-processing successfully	Pass
	Data pre-processing	To verify that data has pre-processed	If invalid data file imported	Invalid detail	Pre-processing unsuccessfully	Pre-processing unsuccessfully	Pass
TC 03	Train the data	To train data	Implement algorithms	Success output	Training successful	Training successful	Pass

TC 04	Detection	Test the trained algorithm	Implement best algorithm to dataset	Success output	Prediction successful	Prediction successful	Pass

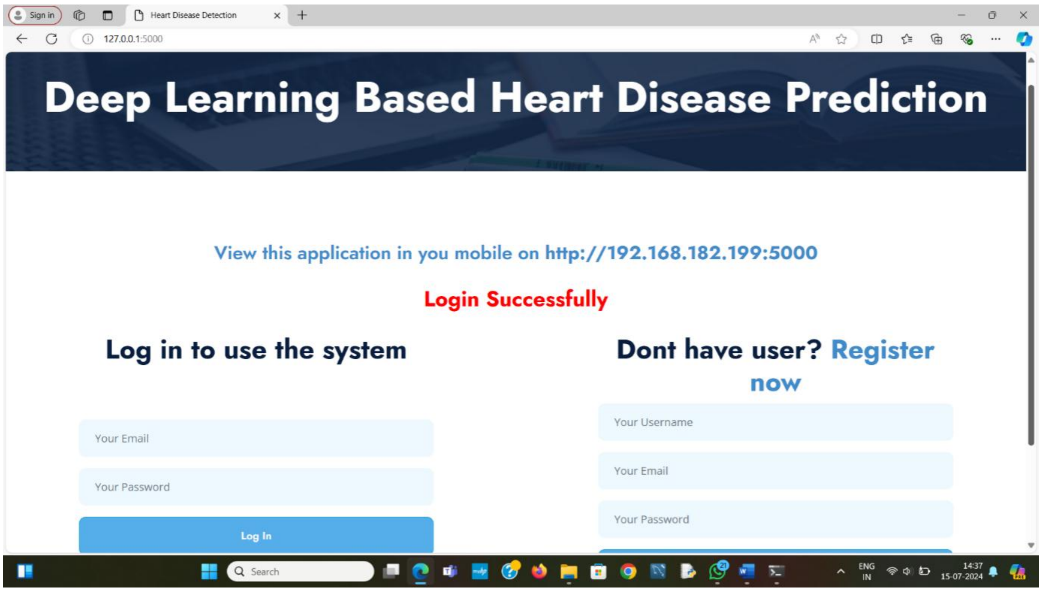
IX. RESULTS

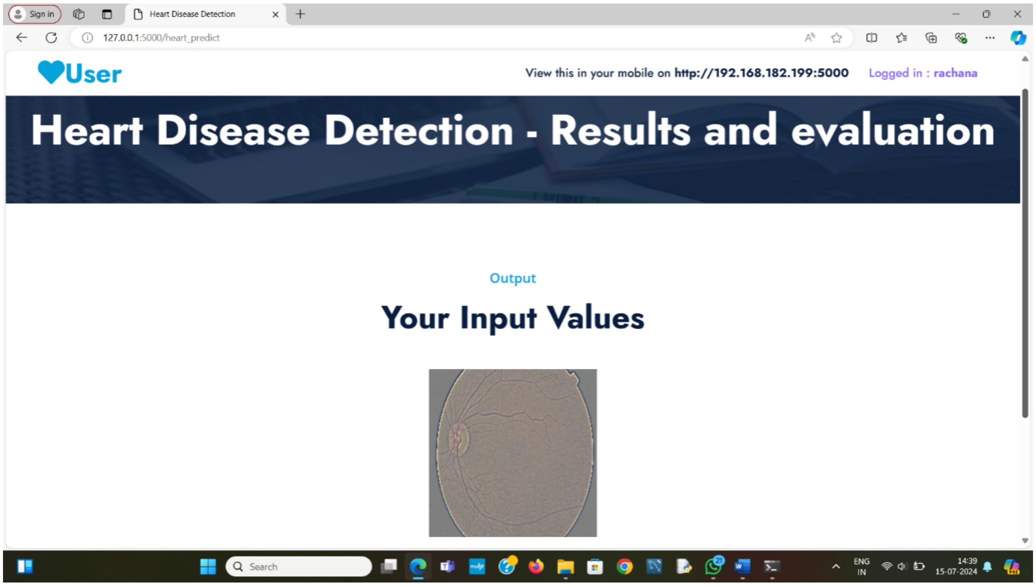


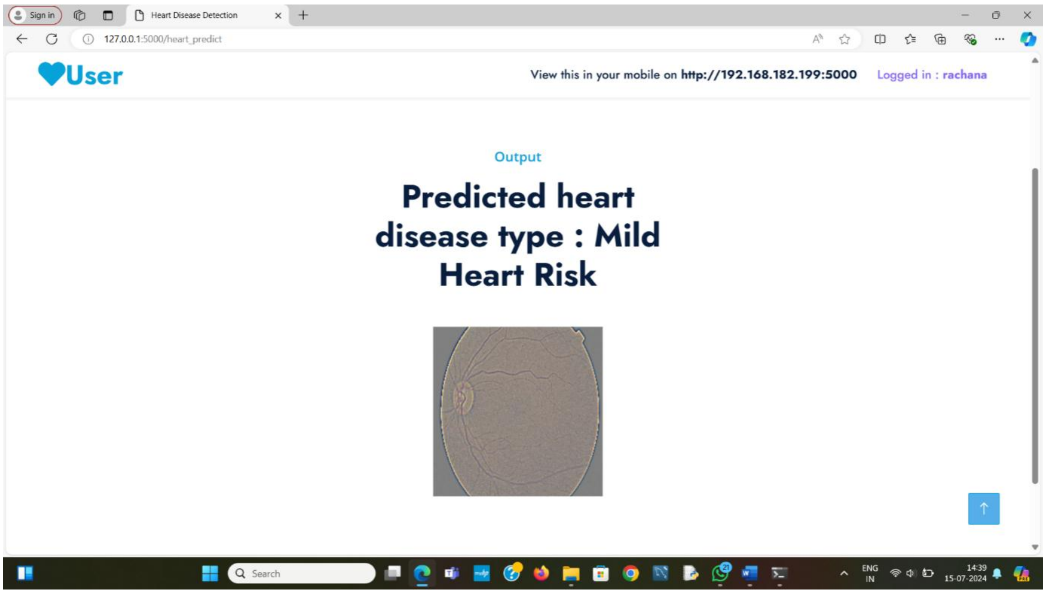


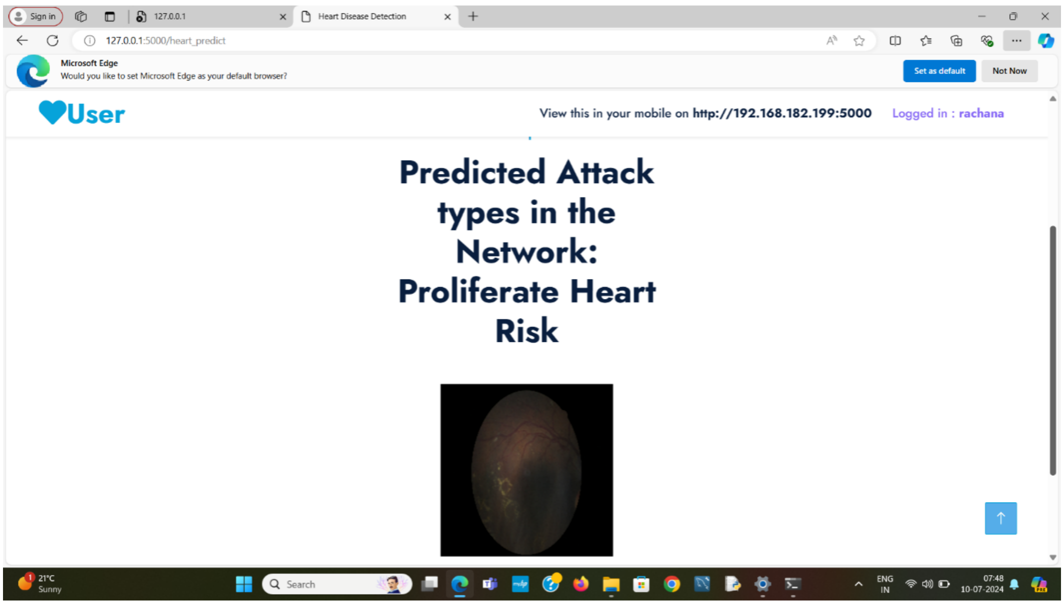

Trainer
View this in your mobile on http://192.168.1.24:5001
Logged in : admin
Dataset
Kinldy copy your dataset to the folder "Dataset" in your project directory with train and test folders
No data found yet
Train The dataset
Apply Deep learning models to train with the Data. Approximate Training time ~ 21 hours
See Console for the training (Skip This step if Already Trained)
Generate Models
Save the trained models in the previous step
Show Accuracy
View Accuracy plot of the data models Saved in previous step
Show Loss
View Loss plot of the data models Saved in previous step

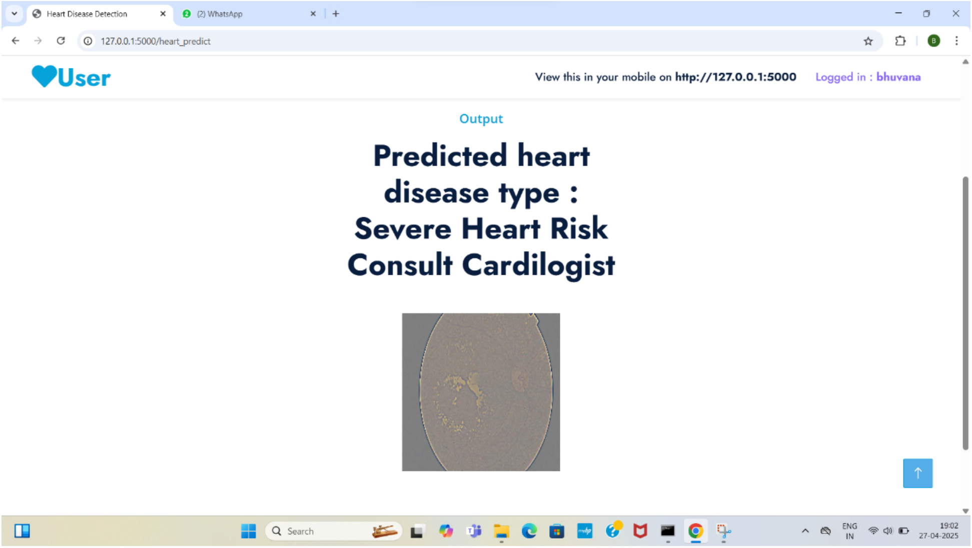

Trainer
View this in your mobile on http://192.168.1.24:5001
Logged in : admin
step if Already Trained)
Accuracy
Loss
© Heart Disease Detection. All Rights Reserved.

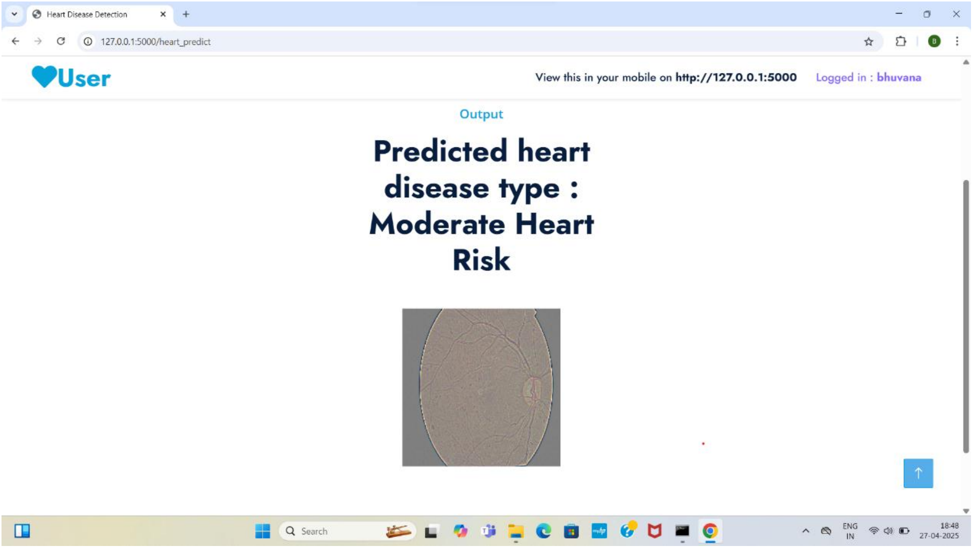

Deep Learning Based Heart Disease Prediction
View this application in you mobile on http://192.168.182.199:5000
Login Successfully
Log in to use the system
Dont have user? Register now
Your Email
Your Password
Log In
Your Username
Your Email
Your Password

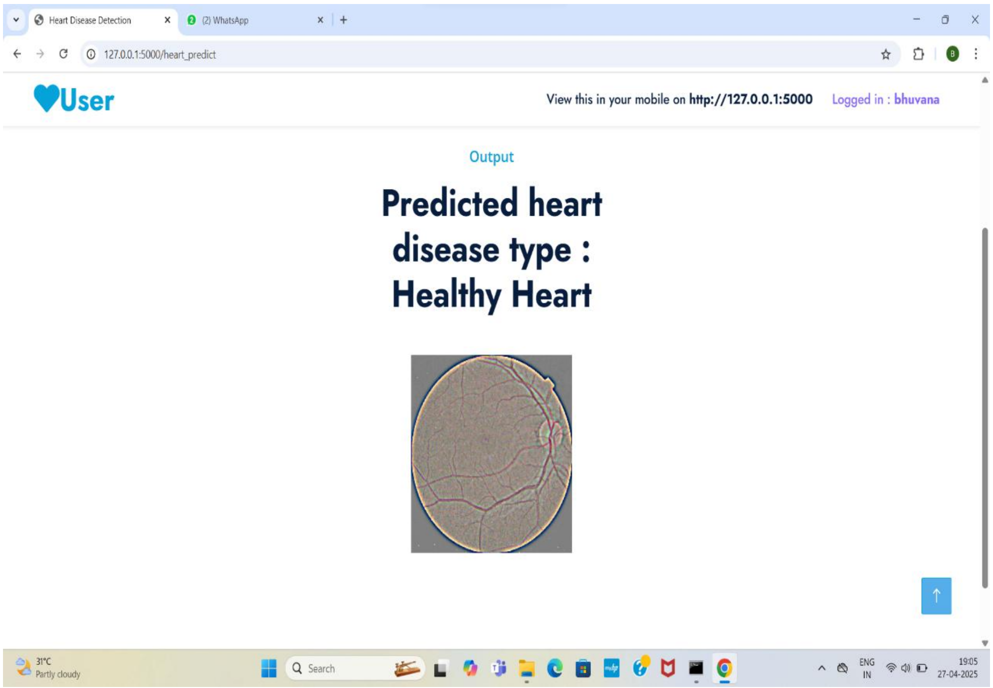

Heart Disease Detection
127.0.0.1:5000/heart_predict
User
View this in your mobile on http://192.168.182.199:5000
Logged in : rachana
Heart Disease Detection - Results and evaluation
Output
Your Input Values


Heart Disease Detection
127.0.0.1:5000/heart_predict
User
View this in your mobile on http://192.168.182.199:5000
Logged in : rachana
Output
Predicted heart disease type : Mild Heart Risk


Heart Disease Detection
127.0.0.1:5000/heart_predict
Microsoft Edge
Would you like to set Microsoft Edge as your default browser?
Set as default
Not Now
User
View this in your mobile on http://192.168.182.199:5000
Logged in : rachana
Predicted Attack types in the Network: Proliferate Heart Risk


Heart Disease Detection
(2) WhatsApp
127.0.0.1:5000/heart_predict
User
View this in your mobile on http://127.0.0.1:5000
Logged in : bhuvana
Output
Predicted heart disease type : Severe Heart Risk Consult Cardilogist


Heart Disease Detection
127.0.0.1:5000/heart_predict
User
View this in your mobile on http://127.0.0.1:5000
Logged in : bhuvana
Output
Predicted heart disease type : Moderate Heart Risk


Heart Disease Detection
(2) WhatsApp
127.0.0.1:5000/heart_predict
User
View this in your mobile on http://127.0.0.1:5000
Logged in : bhuvana
Output
Predicted heart disease type : Healthy Heart

X. CONCLUSION

In conclusion, Recurrent Neural Networks (RNNs) represent a powerful class of neural networks specifically designed for handling sequential data. Their ability to capture temporal dependencies and maintain memory over sequences makes them well-suited for a variety of applications, including the analysis of retinal images for heart disease detection. In the context of cardiovascular health, RNNs can be applied to sequential data derived from medical imaging, time-series physiological measurements, or other relevant sources. In the realm of heart disease detection from retinal images, the application of RNNs opens new possibilities for accurate and dynamic assessments, facilitating early diagnosis and personalized interventions. Continued research and development in this area, coupled with advancements in machine learning techniques, are likely to contribute significantly to the improvement of cardiovascular healthcare outcomes.

XI. FUTURE ENHANCEMENTS

Integrating additional modalities of data can enhance the system's capabilities and provide a more comprehensive view of cardiovascular health.

Potential Additional Modalities

1. Patient Demographics: Incorporating demographic data, such as age, sex, and medical history, can help identify high-risk patients.
2. Clinical History: Integrating clinical history, including previous diagnoses, treatments, and outcomes, can provide valuable context.
3. Genetic Information: Incorporating genetic data can help identify genetic predispositions to cardiovascular disease.
4. Other Medical Imaging Modalities: Integrating data from other imaging modalities, such as:

- Echocardiography: Provides detailed images of heart structure and function.
- MRI Scans: Offers high-resolution images of cardiovascular structures.

Benefits of Multi-Modal Data Fusion

1. Comprehensive View: Combining multiple data sources provides a more comprehensive understanding of cardiovascular health.
2. Improved Accuracy: Fusing multi-modal data can improve disease detection accuracy.
3. Personalized Medicine: Integrating diverse data sources enables personalized treatment plans.

Potential Applications

1. Early Disease Detection: Multi-modal data fusion can facilitate early detection of cardiovascular disease.
2. Risk Stratification: Identifying high-risk patients enables targeted interventions.
3. Treatment Planning: Personalized treatment plans can be developed based on comprehensive patient data.

REFERENCE

[1] Adler ED, Voors AA, Klein L, Macheret F, Braun OO, Urey MA et al (2020) Improving risk prediction in heart failure using machine learning. Eur J Heart Fail 22(1):139–147. https://doi.org/10.1002/EJHF.1628

[2] Akbilgic O, Butler L, Karabayir I, Chang P, Kitzman D, Alonso A et al (2021) Artificial intelligence applied to ecg improves heart failure prediction accuracy. J Am Coll Cardiol 77(18):3045. https://doi.org/10.1016/S0735-1097(21)04400-4

[3] Albert KF, John R, Divyang P, Saleem T, Kevin MT, Carolyn JP et al (2019) Machine learning prediction of response to cardiac resynchronization therapy: improvement versus current guidelines. Circ Arrhythmia Electrophysiol, vol 12(7). https://doi.org/10.1161/CIRCEP.119.007316

[4] Ali MM, Paul BK, Ahmed K, Bui FM, Quinn JMW, Moni MA (2021) Heart disease prediction using supervised machine learning algorithms: performance analysis and comparison. Comput Biol Med 136:104672. https://doi.org/10.1016/J.COMPBIOMED.2021.104672

[5] Araujo M, Pope L, Still S, Yannone C (2021) Prediction of heart disease with machine learning techniques. Graduate Res, Kennesaw State Un

[6] Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324

[7] Caruana R, Karampatziakis N, Yessenalina A (2008) An empirical evaluation of supervised learning in high dimensions. In: Conference: machine learning, proceedings of the twenty-fifth international conference (ICML 2008), Helsinki, Finland

[8] Dalal S, Onyema EM, Kumar P, Maryann DC, Roselyn AO, Obichili MI (2022) A hybrid

machine learning model for timely prediction of breast cancer. Int J Model Simul Sci Comput 0(0):2341023. https://doi.org/10.1142/S1793962323410234

[9] Diwakar M, Tripathi A, Joshi K, Memoria M, Singh P, Kumar N (2021) Latest trends on heart disease prediction using machine learning and image fusion. Mater Today: Proc 37(Part 2):3213–3218.