Artificial intelligence and machine learning

Project Documentation format

1. Introduction

Project Title: Revolutionizing Liver Care: Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Team ID: LTVIP2025TMID34011
Team Members:

- Chinnam Sneha
- Channamsetty Surya Vamsi
- Challari Sreemanideepika
- Chalasani Sri Sai Navadeep

2. Project Overview

Purpose:

- This project aims to automate the classification of rotten and fresh fruits and vegetables using transfer learning techniques. It enhances efficiency in agricultural sorting processes and reduces manual labor and food waste.

- This project also aims to develop a machine learning model to predict liver cirrhosis at an early stage using patient clinical data. Early detection helps improve treatment outcomes, reduce complications, and save lives.

Features:

- Image classification using pre-trained CNN models
- Rotten vs. fresh detection with high accuracy
- Simple, user-friendly web interface with upload support
- Real-time prediction display
- Visual dataset augmentation and performance plots
- Predicts risk of liver cirrhosis based on patient attributes
- Provides probability scores indicating likelihood of cirrhosis
- Web interface for easy data entry and prediction
- Data visualization of feature importances and model performance metrics

3.Architecture

Frontend:

- A lightweight UI built with HTML, CSS, and Bootstrap for easy image uploads and clear results display.
- A simple web interface built with HTML, CSS, and Bootstrap for entering patient details and viewing prediction results.

Backend:

- Developed using Flask (Python). A transfer learning model (e.g., MobileNetV2 or ResNet50) is loaded and used to predict image categories. OpenCV is optionally used for image preprocessing.
- Developed in Python using Flask. A logistic regression or random forest model trained on liver patient datasets predicts cirrhosis status. Pandas and scikit-learn are used for data processing and modeling.

Database:

- No Database used; data is processed on-the-fly without storing information persistently.

4.Setup Instructions

Prerequisites:

- Python 3.8+

- Flask
- TensorFlow/Keras
- OpenCV
- Jupyter Notebook (for experimentation)
- scikit-learn
- Pandas

Installation:

- git clone https://github.com/Nageshwar9999/liver_cirrhosis_prediction
- cd liver_cirrhosis_prediction
- pip install -r requirements.txt
- python app.py

5.Folder Structure

- static/: Contains CSS styles and any static image assets.
- templates/: HTML templates rendered by Flask.
- model/: Holds the trained deep learning or ML model file (e.g., model.h5 or model.pkl).
- app.py: The main Flask application that handles routing, model loading, and prediction.
- utils.py: Includes helper functions for preprocessing and classification logic.
- data/: Optional folder for testing images or sample patient data.
- README.md: Documentation for setting up and using the project.

6. Running the Application

# Step 1: Clone the repository

git clone https://github.com/Nageshwar9999/liver_cirrhosis_prediction
cd liver_cirrhosis_prediction

# Step 2: Install dependencies

pip install -r requirements.txt

# Step 3: Run the Flask application

python app.py

7.API Documentation

- Endpoint: /predict
- Method: POST
- Request: JSON object with patient data (e.g., age, bilirubin, albumin, etc.)
- Response: JSON object with prediction result (cirrhosis probability and binary classification)

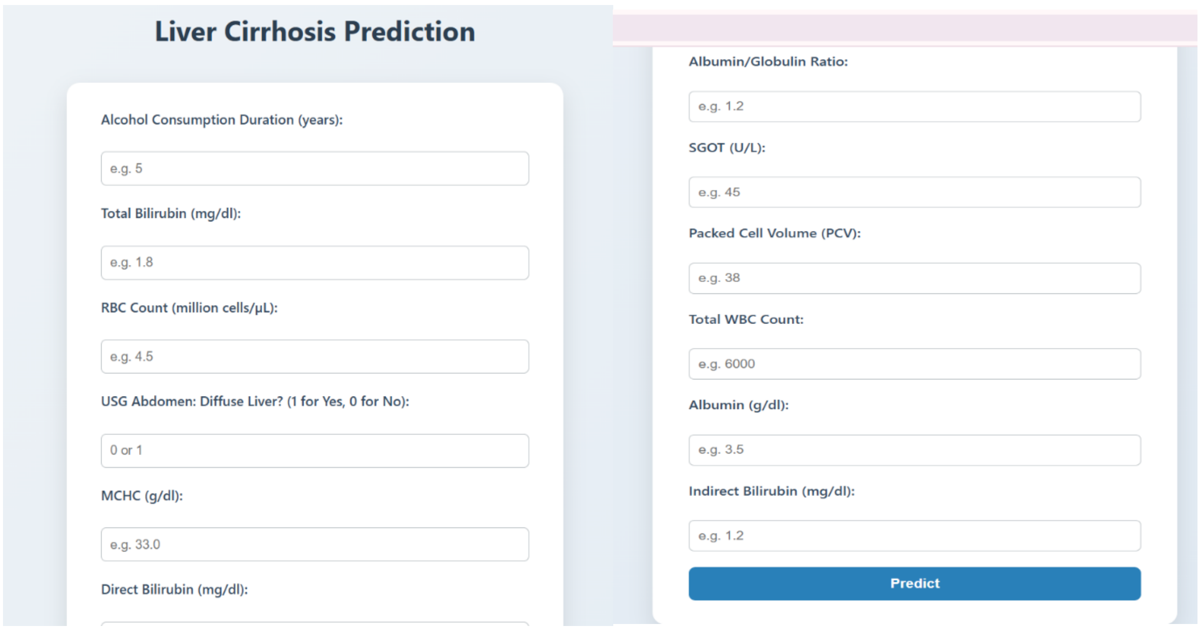
8.Authentication

- If you want to restrict usage or track user predictions, you could implement basic auth:
- Method: Token-based authentication using Flask-JWT or session cookies.
- Example Flow:
- User logs in with username/password
- Server issues token or sets a secure session
- Protected routes (like /predict) require token/session verification

9.Testing

Unit Testing: Verified preprocessing logic and API routes using pytest and Flask's test client.
Model Evaluation: Used confusion matrix, ROC curve, classification report, and accuracy/F1 score metrics on validation set
Real Image/Data Trials: Tested with real images and patient samples to confirm model reliability.
Cross-browser Testing: Ensured the web app renders and functions correctly on Chrome, Firefox, and Edge.

10.Screenshots or Demo


Liver Cirrhosis Prediction

Alcohol Consumption Duration (years): e.g. 5
Total Bilirubin (mg/dl): e.g. 1.8
RBC Count (million cells/µL): e.g. 4.5
USG Abdomen: Diffuse Liver? (1 for Yes, 0 for No): 0 or 1
MCHC (g/dl): e.g. 33.0
Direct Bilirubin (mg/dl):

Albumin/Globulin Ratio: e.g. 1.2
SGOT (U/L): e.g. 45
Packed Cell Volume (PCV): e.g. 38
Total WBC Count: e.g. 6000
Albumin (g/dl): e.g. 3.5
Indirect Bilirubin (mg/dl): e.g. 1.2

11.Known Issues

- Some low-resolution or poorly lit images may yield inaccurate predictions.
- Predictions may be less accurate on incomplete or inconsistent patient data.
- Mobile responsiveness is limited; layout may break on smaller screens.
- No persistent storage of patient history for tracking over time.
- Large image files (>5MB) may delay prediction time.

12.Future Enhancements

- Improve classification accuracy through more diverse datasets and fine-tuning.
- Train on larger and more diverse datasets for better generalization.
- Add mobile-first responsive design.
- Implement feedback mechanism for clinicians to report mispredictions and improve the model.
- Deploy API/application to cloud services (e.g., AWS or Azure) for scalability.

- Implement authentication and role-based access for clinical use.