Liver Disease Detection using Machine Learning

Soham Kokane[1], Aditya Kadu[2], Sakshi Bhalerao[3], Rohini Bhorde[4], Prof. Anuradha Kukarni [5]
[1, 2, 3, 4, 5] *Department of Information Technology, Sinhgad Institute of Technology, Lonavala*

Abstract - Liver is a very essential organ in the human body. It is essential to recognize or diagnose the disease early. This considerably aids in the early avoidance of disease with minimal medication. Conventional methods include Liver Function Tests and test results. Early detection of liver disease is very difficult. This is because symptoms of the disease become apparent only in the later stages of the disease. This system of Machine learning facilitates early disease detection, Identifying elements that lead to fatal liver impairment. Predicting the disease in its early stages is a difficult task for doctors and scientists due to the apparently sensitive signs. Effects will become apparent only when it is too late. The initiative seeks to use machine learning techniques to address this issue and improve the victims of the disease. Because there are few signs of liver disease, it is difficult to diagnose and symptoms usually do not appear until it is too late. The aim is to study and use a classification approach to distinguish between liver disease and healthy individuals, if diseased then further classified into the level of disease and its type. Also, precautions are provided for any symptoms. Consequently, ML techniques have identified liver disease in individuals.

I. INTRODUCTION

Liver diseases are becoming more common than ever due to the increasingly sedentary lifestyle trend and lack of physical activity. The intensity is still controllable in rural areas, but liver disease is increasingly very prevalent in urban areas, particularly metropolitan areas. Prediction of Liver Diseases is necessary for Human beings' livelihood. The liver is the largest internal Organ of the human body and when it is injured, the human being's life is automatic gets troubled. The different types of liver diseases identified in Human beings are fatty liver, cirrhosis, liver cancer, hepatitis, liver tumor, etc. The percentage of diseases is increasing due to the increase in the consumption of alcohol, Drugs, pickles, and food. Each year Millions of people die due to liver diseases. The liver can be affected by several disease states. When alcohol is used for long periods of time, it changes the liver's metabolism, which may have overall bad impacts. hemochromatosis causes liver problems.

A group of blood tests called liver function tests can be used to diagnose liver disease. Other blood tests can be done to look for specific liver problems or genetic conditions. Imaging tests. An ultrasound, CT scan and MRI can show liver damage

II. RELATED WORK

OVERVIEW

Liver diseases are becoming more common than ever due to the increasingly sedentary lifestyle trend and lack of physical activity. The intensity is still controllable in rural areas, but liver disease is increasingly very prevalent in urban areas, particularly metropolitan areas. Prediction of Liver Diseases is necessary for Human beings' livelihood. The liver is the largest internal Organ of the human body and when it is injured, the human being's life is automatic gets troubled. The different types of liver diseases identified in Human beings are fatty liver, cirrhosis, liver cancer, hepatitis, liver tumor, etc. The percentage of diseases is increasing due to the increase in the consumption of alcohol, Drugs, pickles, and food. Each year Millions of people die due to liver diseases. The liver can be affected by several disease states. When alcohol is used for long periods of time, it changes the liver's metabolism, which may have overall bad impacts. hemochromatosis causes liver problems.

EXTERNAL INTERFACE REQUIREMENT

User Interface

Application Based Alzheimer disease detection

SOFTWARE REQUIREMENTS:
Operating System: Windows 10
IDE: Pycharm, Spyder

Programming Language: Python
HARDWARE REQUIREMENTS:
RAM : 8 GB
As we are using Machine Learning Algorithm and Various High Level Libraries
Laptop
RAM minimum required is 8 GB.
Hard Disk : 40 GB
Data Set of CT Scan images is to be used hence minimum 40 GB Hard Disk memory
is required.
Processor : Intel i5 Processor
Pycharm IDE that Integrated Development Environment is to be used and data loading should be fast hence Fast Processor is required
IDE : Pycharm
Best Integrated Development Environment as it gives possible suggestions at the
time of typing code snippets that makes typing feasible and fast.
Coding Language : Python Version 3.5
Highly specified Programming Language for Machine Learning because of availability of High Performance Libraries.
Operating System : Windows 10
Latest Operating System that supports all type of installation and development Environment

NON- FUNCTIONAL REQUIREMENT

Performance Requirements: The performance of the functions and every module must be well. The overall performance of the software will enable the users to work eciently. Performance of encryption of data should be fast. Performance of the providing virtual environment should be fast Safety Requirement The application is designed in modules where errors can be detected and xed easily. This makes it easier to install and update new functionality if required.

Safety Requirement: The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required

Software Quality Attributes:
Our software has many quality attribute that are given below:-
Adaptability: This software is adaptable by all users.
Availability: This software is freely available to all users. The availability of the
software is easy for everyone.
Maintainability: After the deployment of the project if any error occurs then it can
be easily maintained by the software developer.
Reliability: The performance of the software is better which will increase the reliabilityof the Software.
User Friendliness: Since, the software is a GUI application; the output generated is
much user friendly in its behavior.
Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
Security: Users are authenticated using many security phases so reliable security is
provided.
Testability: The software will be tested considering all the aspects.

SOFTWARE INFORMATION

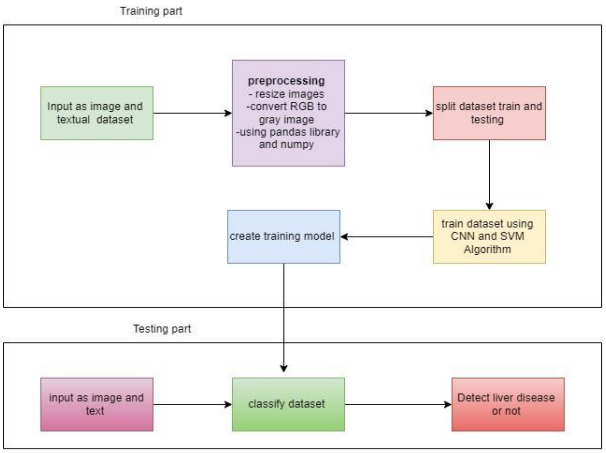
Python is an interpreted, high-level and general-purpose programming language.
Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its
language constructs and object-oriented approach aim to help programmers write
clear, logical code for small and large-scale projects.
Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented,
and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.
Python was created in the late 1980s as a successor to the ABC language.
Python 2.0, released in 2000, introduced features like list comprehensions and a
garbage collection system with reference counting.
Python 3.0, released in 2008, was a major revision of the language that is not
completely backward-compatible, and much Python 2 code does not run unmodified
on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for
2015), and ”Python 2.7.18 is the last Python 2.7 release and therefore the last Python
2 release.”[30] No more security patches or other improvements will be released for
it. With Python 2’s end-of-life, only Python 3.6.x and later are supported.
Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source
reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.
Python was conceived in the late 1980s by Guido van Rossum at Centrum
Wiskunde Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with
the Amoeba operating system. Its implementation began in December 1989. Van
Rossum shouldered sole responsibility for the project, as the lead developer, until 12
July 2018, when he announced his ”permanent vacation” from his responsibilities
as Python’s Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project’s chief decision-maker.
He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry
Warsaw, Carol Willing and Van Rossum to a five-member ”Steering Council” to lead the project

Anaconda: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free. Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages[citation needed]. It will install a package and any of its dependencies regardless of the state of the existing installation[citation needed]. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail. In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done. Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user’s own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda

environment using pip, and conda will keep track of what it has installed itself and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda

SYSTEM ARCHITECTURE



I. Fig. SYSTEM ARCHITECTURE

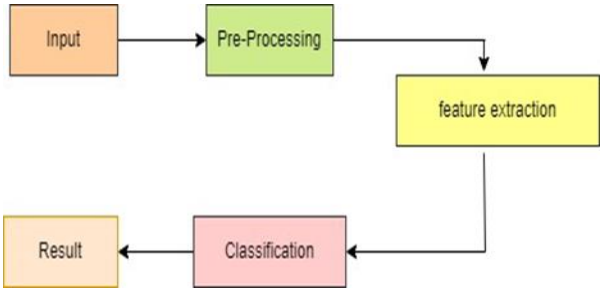


Fig. Data Flow diagram

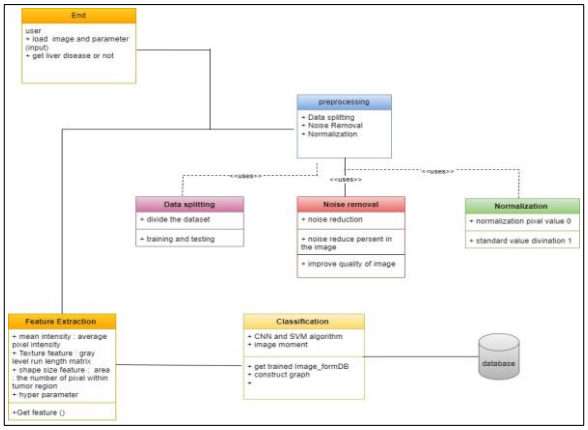
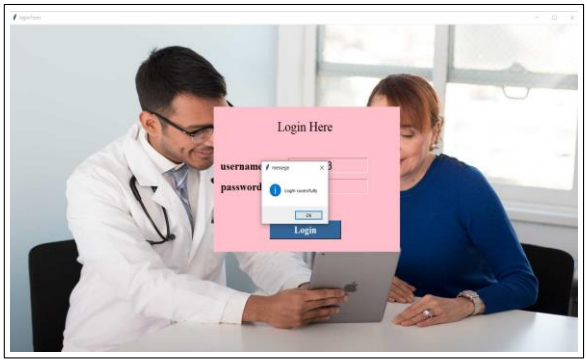


Fig. class diagram

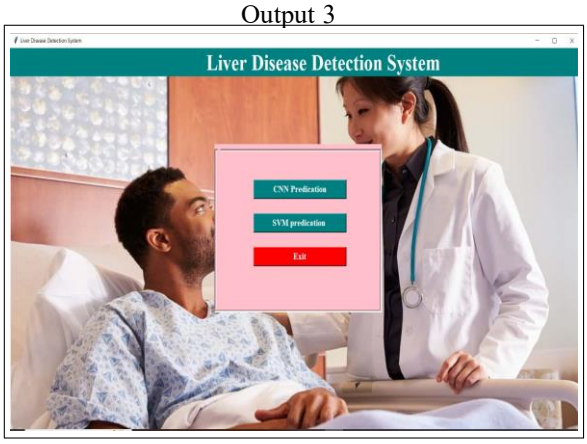
II. IMPLEMENTATION AND RESULTS

Output 1

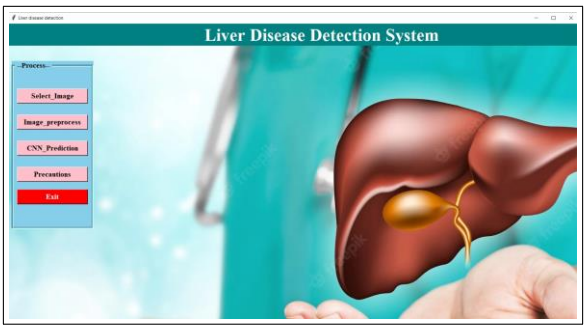
Output 2



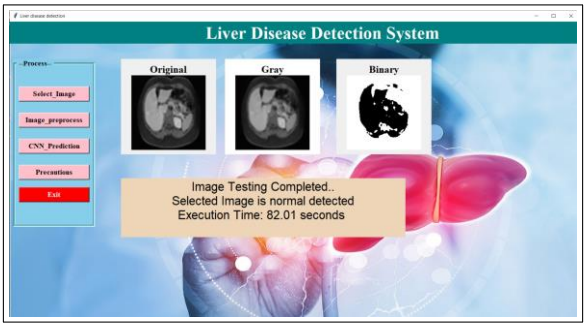
Output 3



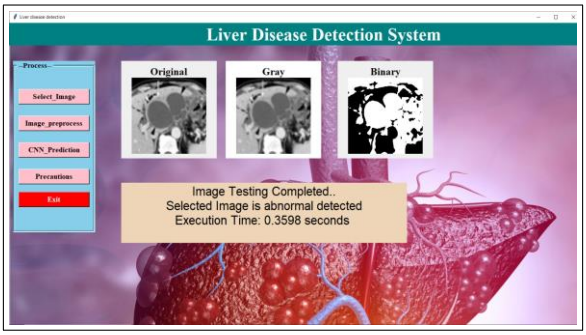
Output 4



Output 5

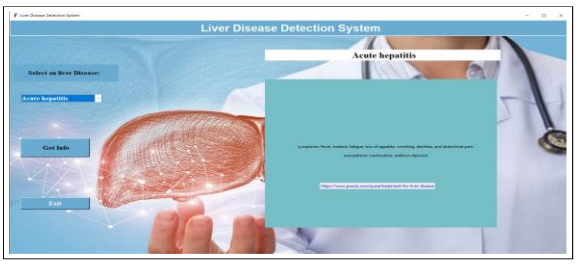


Output 6

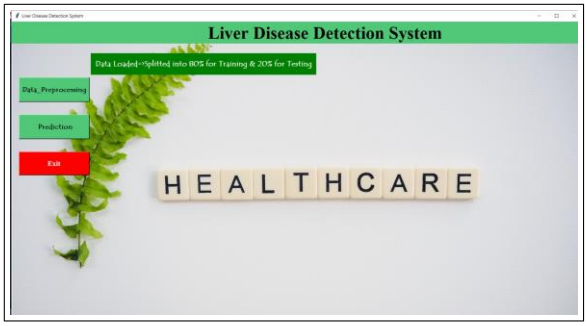


Output 7

Output 8



Output 9



Output 10

Output 11

Output 12

CONCLUSION

These techniques have revealed the variations in the texture patterns of normal and diseased livers. These results were a significant achievement in confirming the variation in liver texture from disease to disease, and even from mild to severe cases in a single disease.

These methods are simpler to implement and easy to handle. We cannot speak of the perfection of a single method. The analysis can be varied from disease to disease. However, the combination of different techniques can provide promising results when discriminating different tissue textures.

REFERENCES

[1] Image processing tool for detection of liver disease,Juan Rodolfo MaestreRendon ,Norma Ram´ırez Hernandez,978-1-6654-1262-9/21/31.00 ©2021 IEEE

[2] Convolutional Neural Network Architectures for Fatty Liver Disease Detection,Shaun Almeida, Hirtika Mirghani, Maniroja Edinburgh,978-1-6654-6539- 7/22/31.00 ©2022 IEEE

[3] Liver disease detection using machine learning techniques,Deepika Bhupathi1 , Christine Nya-Ling Tan1 , Sreenivas Sremath Tirumula1 and Sayan Kumar Ray1, 27 September 2022

[4] Liver Disease detection due to excessive alcoholism using Data Mining Techniques,1 Insha Arshad,2Chiranjit Dutta,3Tanupriya Choudhury,4Abha Thakral,978-1-5386-4485-0/18/31.00 ©2018 IEEE

[5] Measuring Steatosis in Liver Biopsies using Machine Learning and Morphological Imaging ,Nikolaos Giannakeas, Markos G. Tsipouras,063-7125/17 31.00 © 2017 IEEE DOI 10.1109/CBMS.2017.98 2372-9198/17 31.00 © 2017 IEEE

[6] Diagnosis of Liver Disease using Machine Learning Models, Baddigam Jaya Krishna Reddy,978-1-7281-5464-0/20/31.00 ©2020 IEEE

[7] Liver Disease Detection: Evaluation of Machine Learning Algorithms Performances With Optimal Thresholds,Aritra Pan, Indian Institute of Management, Bodh Gaya, India*, 24 October 2022

[8] Deep Convolutional Neural Network Based Analysis of Liver Tissues Using Computed Tomography Images,Mehrun Nisa 1,2, Saeed Ahmad Buzdar 2, Khalil Khan 3,* and Muhammad Saeed Ahmad 4,Symmetry 2022

[9] Prediction and Detection of Liver Diseases using Machine Learning, R.T. Umbare, Omkar Ashtekar,979-8-3503-9781-9/23/31.00 ©2023 IEEE

[10] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, “Early diagnosis of alzheimer’s disease with deep learning,” in Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on. IEEE, 2014, pp. 1015– 1018.