Journal of
Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.9854909

CNN BASED DIABETIC RETINOPATHY DIAGNOSIS ON RETINAL FUNDUS IMAGES

T.Papitha Christobel
Assistant Professor, Department of MCA, SRM Institute of Science and Technology, Ramapuram, Chennai, email id : papithat@srmist.edu.in

S. Meenakshi
Assistant Professor, Department of MCA, SRM Institute of Science and Technology, Ramapuram, Chennai, email id:meenakss3@srmist.edu.in

N. Indumathi
Assistant Professor, Department of MCA, SRM Institute of Science and Technology, Ramapuram, Chennai, email id: induarivu27@gmail.com

Abstract
Diabetic Retinopathy is a severe diabetes complication that can cause vision loss or blindness. Image analysis has been increasingly used by medical researchers to accurately diagnose diseases. As a result, a computational model was developed to predict the presence of Diabetic Retinopathy (DR) using retinal images and a neural network. The computational model consists of two stages: feature extraction and classification. In the feature extraction phase, digital fundus images were analyzed to identify the most relevant features such as Blood Vessels and Micro aneurysms. The study was conducted using data from the Diabetic Retinopathy dataset available on Kaggle Community. Our objective is to use deep neural networks to automatically differentiate between healthy and pathological retinal fundus images. This is because deep learning is a highly effective machine learning method that has been proven to be extremely accurate in various computer vision tasks. To achieve this goal, we have employed convolutional neural networks (CNN) in our study to classify retinal images as either healthy or unhealthy. Finally, a convolutional neural network (CNN) was used to predict Diabetic Retinopathy. The proposed methodology achieved a 95.41% accuracy rate based on the model's results.
Keywords: Diabetic Retinopathy, Retinal Images, Digital Fundus Images, Blood Vessels, Micro Aneurysms, CNN and Deep Learning

I. INTRODUCTION

Diabetic Retinopathy (DR) is a well-known eye disease that affects a significant number of people globally, especially those who have had diabetes for an extended period. It is a leading cause of visual impairment, and the World Health Organization predicts that diabetes will cause approximately 1.6 million deaths by 2020. In the United States, 40% of diabetic patients, out of a total of 29.1 million, had DR at various stages in 2018, and the number of cases is increasing. DR is divided into two types: non-proliferative (NPDR) and proliferative (PDR). The initial phase of DR, NPDR, is characterized by increased vascular penetrability and narrowed impediment in the retinal vasculature. Timely detection of DR is critical to ensure prompt treatment of the patient, and recent advancements in computer technology have made it possible to use devices for the automatic detection of DR. The Automated DR conclusion framework, based on profound learning innovations, has demonstrated significant accuracy in recognizing and analyzing DR.

II. MATLAB

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi- domain simulation and Model-Based Design for dynamic and embedded systems. MATLAB is a wonderful environment for serious numerical computations as well as for graphics. It is replacing FORTRAN and other languages that have often been used for numerical scientific computations. MATLAB has several options for on-line assistance. MATLAB offers a tutorial, which can be accessed from the Help menu or by typing 'demo' at the command prompt. It would be a good idea to run through some of these demos to get an idea of how MATLAB does "stuff"! The index of MATLAB help information can be accessed from Help -> MATLAB Help. Here you can find information on getting started, using MATLAB, and implementing built-in functions. Information on the built-in functions may also be obtained by using the commands 'help' and 'look for' in the command window. 'Help function name' provides help on the function if you know its exact name. If you don't know the exact name of the function, use 'look for keyword' to get a list of functions with string keyword in their description. A MATLAB program, called an M-file, is just a list of MATLAB commands, the same commands that you can use interactively in the command window. You can write that list of commands in a text file and then execute the program from the command window. To create an M-file, start MATLAB and under the File menu select New and M-file. Type your commands in the M-file window. Store the M-file with the same name as the function, and with the suffix ".M" such as "Prog.M". The file can be stored and run from a floppy or the hard disk. After you have changed an M-file, remember to save it before using the function or script. MATLAB uses the saved version of the program, and not the version displayed in the window. MATLAB programs in M-files can be classified into two groups: script files and function files. They differ in two things: (i) the way you execute them, and (ii) the type of variables they involve.

A. SCRIPT FILES

Script files are M-files that can be executed by typing their names in the command window, or calling them from other M-files. The variables they contain or define are global variables. That is, after you execute a script file all variables involved would be in memory and usable from the command window.

B. Using MATLAB

The best way to learn to use MATLAB is to sit down and try to use it. In this handout are a few examples of basic MATLAB operations, but after you've gone through this tutorial you will probably want to learn more. Check out the "Other Resources" listed at the end of this handout. The Beginning When you start MATLAB, the command prompt ">>" appears. You will tell MATLAB what to do by typing commands at the prompt.

C. CREATING MATRICES

The basic data element in MATLAB is a matrix. A scalar in MATLAB is a 1x1 matrix, and a vector is a 1xn (or nx1) matrix.

D. ADVANCED OPERATIONS

There's a lot more that you can do with MATLAB than is listed in this handout. Check out the MATLAB help or one of the "Other Resources" if you want to learn more about the following more advanced tools:

- Numerical integration (quad)
- Discrete Fourier transforms (fft, ifft)
- Statistics (mean, median, std, var)
- Curve fitting (cftool)
- Signal processing (sptool)
- Numerical integration of systems of ODEs (ode45)

E. CREATING MATRICES:

- zeros(m, n): matrix with all zeros
- Ones (m, n): matrix with all ones.
- eye(m, n): the identity matrix
- rand(m, n): uniformly distributed random
- randn(m, n): normally distributed random
- magic(m): square matrix whose elements have the same sum, along the row, column and diagonal.

III. SYSTEM TESTING

A software program consists of logical components of a system that must accurately compile, test, and integrate with other programs to function successfully. It is the responsibility of the programmer to develop a program that is free of errors. During program testing, syntax and logic errors are detected. When there is a discrepancy between the actual output and the intended result, the sequence of instructions must be traced to identify the problem. To isolate the issue, the program values are compared against desk-calculated values. Testing is a crucial stage in the software development life cycle (SDLC).

3.1 TYPES OF TESTING

3.1.1 Unit Testing

Unit testing involves the design of test cases that validates that the internal program logic is functioning properly, and the program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.1 Integration Testing

Integration testing can proceed in a number of different ways, which can be broadly characterized as top down or bottom up. On top-down integration testing the high-level control routines are tested first, possibly with the middle level control structures present only as stubs.

6.1.2 Functional Testing

Functional testing is a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (Not like in white-box testing).

6.1.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integrations points.

S.NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Search whether Matlab isvisible or not	Visible	Pass	Pass
2	Check whether the signicon is clickable	Clickable	Pass	Pass
3	Check whether the email already account created	Email entered	Pass	fail
4	Check whether the account is accept specialcharacter as password	Password accepted	Pass	Pass
5	Check whether the application is installed ornot	Application Installed	Pass	Pass
6	Check whether the sub directories application isdownloaded or not	Downloaded	Pass	Pass
7	Clicking on desktop iconmust open it	Application opened	Pass	Pass
8	Check whether the open option is working to openthe coding file.	Coding file opened	Pass	Pass
9	Check whether the coding is running or not	Running	Pass	Pass
10	Input the retina image	Stored successfully	Pass	Pass
11	While run the program retina output show 5 retina image	Running successfully	Pass	Pass
12	The accuracy most not come in special character	Not coming in special character	Pass	Pass
13	Input the other image	Error occur	Pass	Pass

Table 1. Test Cases

IV. EXISTING SYSTEM

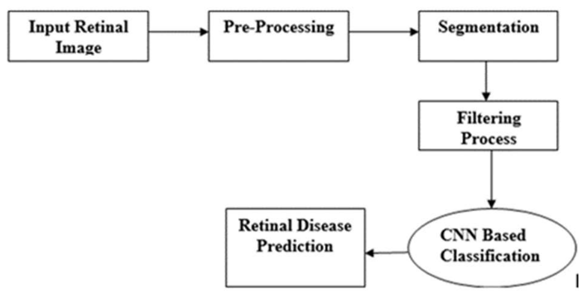
Diabetic retinopathy is divided into two types: non-proliferative (NPDR) and proliferative (PDR). In the initial stage, NPDR is characterized by increased vascular permeability and a narrowing of the retinal blood vessels. Fundus photography is an effective diagnostic tool for identifying retinal conditions such as aneurysms, hemorrhages, and hard exudates, which makes it useful in the early detection of DR. It is critical to detect the disease early to provide prompt treatment to the patient. Thus, regular monitoring of the retina is essential, and with advancements in computer technology for vision, it can be utilized as a tool for natural detection. Drawbacks of existing system are the recognition of fundus issues is deemed to be intricate and the precision of the recognition has diminished.

V. PROPOSED SYSTEM

Various approaches have been proposed for detecting DR disease. Among them, the Automated DR diagnosis system that employs deep learning technologies has demonstrated greater accuracy than other computer vision techniques. This section aims to provide an understanding of the suggested model framework and analysis materials, along with the development of convolutional neural networks. Benefits of proposed system High prediction accuracy and It is an open source.

4.1 DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.



Fig, 1.Data flow diagram

4.2 USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

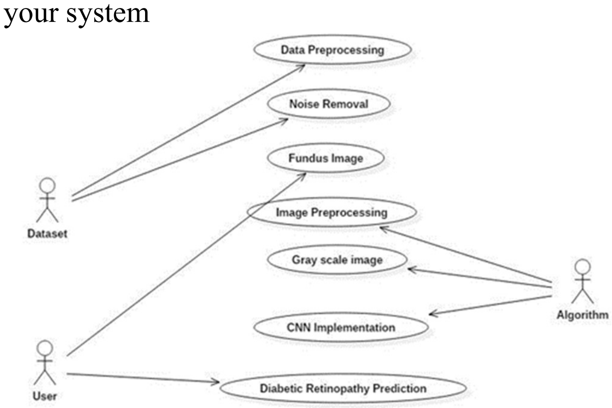


Fig. 2. Use case diagram

4.3 ARCHITECTURE DIAGRAM

An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element. This diagram gives a top-level view of software's structure. To elaborate, it generally includes various components that interact with each other and how the software interacts with external databases and servers. It's useful for explaining software to clients and stakeholders; and assessing the impact of adding new features or upgrading, replacing, or merging existing applications.

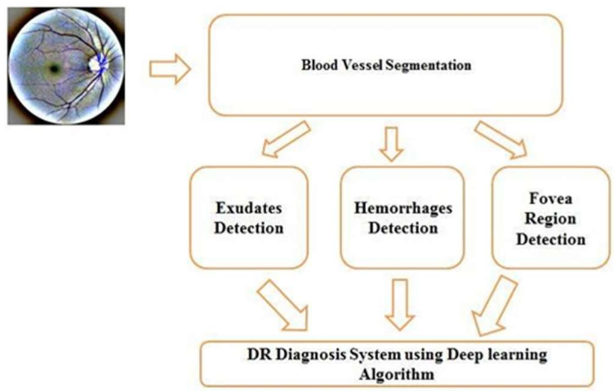


Fig. 3. Architecture diagram

4.4 ABOUT MODULES

- **Fundus input image**

This module is used to input the image for segmentation.

- **Image Preprocessing**

This component aims to preprocess the image for segmentation. It is important to perform the image preprocessing step to address issues caused by variations in the image and to normalize the image. Initially, the pixel values of the images should be rescaled to a value between 0 and 1.

- **Image based Segmentation**

The proposed approach employs an automated and unsupervised method for segmenting vessels in retinal fundus images.

- **Evaluation of Performance**

The proposed approach has been evaluated based on various metrics, including accuracy and precision, to assess its performance.

4.5 IMPLEMENTATION

Obtain trustworthy information on the retina, such as diabetic retinopathy, from reliable sources like government websites. Perform data cleaning and preprocessing to remove any missing or incorrect data and use methods to convert categorical data into numerical data. Create additional features based on domain expertise or statistical data analysis to improve the prediction potential of the model. For instance, you can incorporate new features like the time of day or the duration of diabetes.

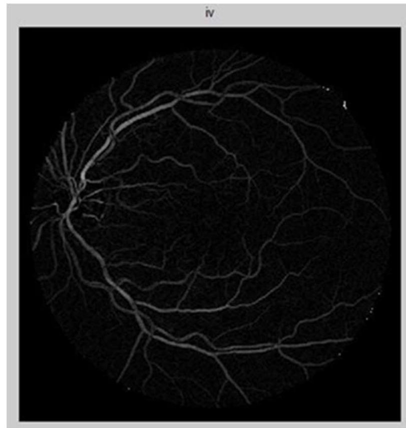


Fig. 4. Input Image

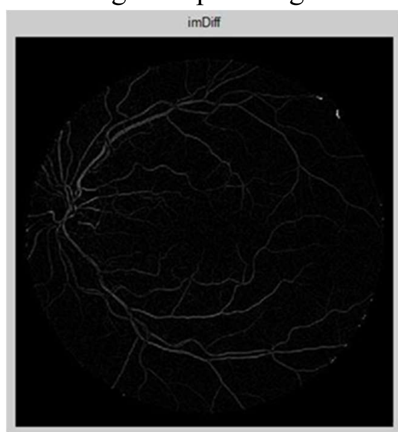


Fig. 5. Smoothness

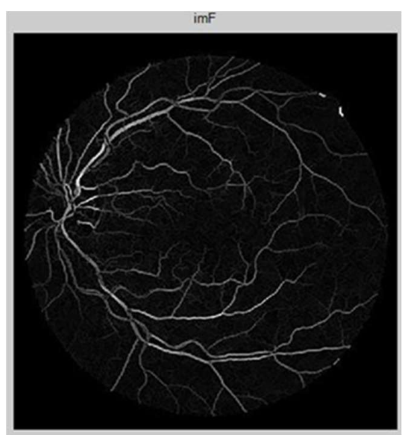


Fig. 6. Segmentations

Fig. 7. Image Coefficient

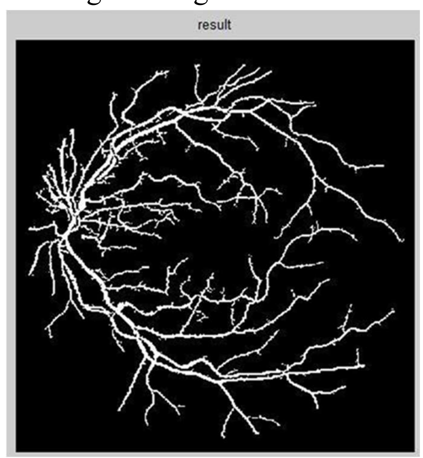


Fig. 8. Blood Vessel

VI. CONCLUSION

Diabetic Retinopathy is a common cause of blindness, and early detection and diagnosis are crucial for patients. However, identifying the condition from retinal images can be challenging for ophthalmologists. Various approaches based on image processing have been employed for DR detection. This research presents a new identification method based on digital image processing for extracting fundus images from damaged retinas. The MATLAB program is used to analyze and classify the fundus images into four categories: exudates, micro aneurysms, optical discs, and hemorrhages. The proposed method also considers the proliferative and non-proliferative stages of DR. The morphological operations of erosion and dilation are employed to learn the fundus's positioning pattern and identify four eye abnormalities. Several performance analyses support the proposed approach, demonstrating 98% accuracy in detecting PDR and NPDR within 39 seconds. The scope of the project is wide-ranging, and we aim to make it flexible and adaptable for use in various settings. It is crucial to improve the de-noising stage to address the mentioned limitations effectively. Additionally, to extend the concept, we need to incorporate new datasets that cover all types of alterations that may occur in retinal images due to DR. This will ensure that we do not assume perfect conditions in our analysis.

REFERENCE

[1] Dutta, S., Manideep, B.C., Basha, S.M., Caytiles, R.D. and Iyengar, N.C.S., 2018. Classification of Diabetic Retinopathy Images by Using Deep Learning Models. International Journal of Grid and Distributed Computing, 11(1), pp.89-106.

[2] Gao, Z., Li, J., Guo, J., Chen, Y., Yi, Z. and Zhong, J., 2018. Diagnosis of Diabetic Retinopathy Using Deep Neural Networks. IEEE Access, 7,pp.3360-3370.

[3] T.Papitha Christobel, A.Sasi Kumar, "Predictive Analysis in Intelligent Healthcare Framework Using Big Data Applications",International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019.

[4] Mohammadian, S., Karsaz, A. and Roshan, Y.M., 2017, November. Comparative Study of Fine- Tuning of Pre-Trained Convolutional Neural Networks for Diabetic Retinopathy Screening. In 2017 24th National and 2nd International Iranian Conference on Biomedical Engineering (ICBME) (pp. 1-6).IEEE.

[5] Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J. and Kim, R., 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama, 316(22),pp.2402- 2410.

[6] Asiri, N., Hussain, M. and Abualsamh, H.A., 2018. Deep Learning based Computer-Aided Diagnosis Systems for Diabetic Retinopathy: A Survey. arXiv preprintarXiv:1811.01238.

[7] Kumar, S. and Kumar, B., 2018, February. Diabetic Retinopathy Detection by Extracting Area and Number of Microaneurysm from Colour Fundus Image. In 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 359-364). IEEE.

[8] Faust, O., Acharya, R., Ng, E.Y.K., Ng, K.H. and Suri, J.S., 2012. Algorithms for the automated detection of diabetic retinopathy using digital fundus images: a review. Journal of medical systems, 36(1),pp.145-157.

[9] T.Papitha Christobel, A.Sasi Kumar, "Pregnancy Period Diabetic and Blood Pressure Predictive Analysis using HCNN-LSTM" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 –8958,Volume-9Issue-3, February 2020.