Severity Detection of Diabetic Retinopathy Using Deep Learning Algorithms

ANUSHA.M, MANASA.M, PUNITH.H.N, RAHUL.V.SANKA, SARATH.KUMAR.V

ABSTRACT There are several deep literacy ways that are used to perform the prophetic analytics over big data in colorful medical tasks. Prophetic analytics in medical healthcare is a grueling task yet eventually helping the interpreters handle big data- informed timely opinions about case's medical health and treatment. This paper discusses the prophetic analytics in healthcare. Case's medical record is attained for experimental exploration. The two infrastructures of deep literacy are enforced. Performance and delicacy of these applied algorithms are enforced and compared. Different deep literacy ways used in this exploration that reveals which algorithm is best suited for the vaticination of diabetes over the case. This paper aims to help croakers and interpreters in early stage to prognosticate diabetic retinopathy using deep literacy ways. Keywords Dataset, ResNet, Convolutional Neural Network.

INTRODUCTION

The medical industry is a vast sensitive metadata that needs careful handling. One of the fast growing deadly diseases across the globe is Diabetes Mellitus. A reliable prediction system is needed by medical practitioners to diagnose this ailment. There are some useful deep learning methods for data analytics which provide rich information from different perspectives and put it into concise representations. The capability to access and process large data sets can be very insightful, but only if some particular data mining techniques are applied to them.

The whole purpose of this is to establish new patterns, explain these patterns in order for them to make sense and give out concrete vital and helpful facts throughout the course of examination. Heart disease, kidney disease, nerve damage and blindness are illnesses that may result from diabetes.

Diabetes data mining is important for major issues efficiently. The methods used by the researchers to mine the diabetes dataset are usually discovered through effective means such as using efficient algorithms or any other technique by which they can identify specific characteristics about diabetes dataset.

In this study, medical bioinformatics analyses are carried out in predicting diabetes mellitus. Diabetic diagnosis employs WEKA software as a mining tool. The Pima Indian diabetes Data base was acquired from UCI repository that used for future analysis. Research and analyze the dataset to construct a good model which will be able to predict and diagnose diabetic diseases. Diabetic retinopathy is one of the most common complications of diabetes caused by changes in blood vessels within the retina, and it's among the leading causes of blindness in developed world. However, ophthalmologists still manually screen for diabetic retinopathy; a time-consuming process therefore this paper seeks at automating diagnosis of different stages of the illness through deep learning. In our approach, we trained a Deep Convolutional Neural Network model on an outsized dataset consisting of around 35,000 images to automatically diagnose and thereby classify high-resolution fundus images of retina into five stages supported their severity. This research builds an application system that would take patient details as input parameters alongside with image fundus features of his or her eyes.

The deep learning convolutional neural networ k model will add features of the fundus image and then get the output with the help of operati ons such as Relu and softmax and optimization techniques such as Adam. The results obtaine d from the convolutional neural network (CNN
) model will create a joint report together with the patient data. In this study, we aimed to use similar implementation methods to improve ac curacy and then compare their performance usi ng ResNet, CNN. The goal of machine learnin g is to predict long-
term events from historical data. Machine lear

ning focuses on changing situations when com puters are exposed to new data; hence the basi cs of machine learning is the use of simple ma chine learning algorithms using Python. It pro vides training data to the algorithm

so that the algorithm uses the training data to provide pre dictions for changing data.
Machine learning broadly falls into three categ ories. Study guides provide data entry and mat ching text to find previously typed data. There are no written notes of the neglected education
. It feeds directly into the learning algorithm. T he algorithm must find a set of input data. Fina lly, the learning support dynamically interacts with its environment and receives suggestions to improve its work. Deep learning is a branch of artificial intelligence in which networks are adapted to learn from anonymous or unsupervi sed data. This technique is also called deep neural learning or deep neural networks. Deep learning consists of a technique called convolutional neural network (ConvNet/CNN) that takes input images and assigns importance (learned weights and biases) to various features/objects in the image. ConvNet requires more prioritization than other classification algorithms. In the first method of deep learning, filters are created manually and with sufficient training, convolutional networks can learn the filters/features.

Diabetic retinopathy is a medical problem that occurs as a result of damage to the blood vessels in the retina, the tissue at the back of the eye, and can lead to blindness and various eye problems, depending on the severity of diabetic retinopathy. It has been found that 40% to 45% of people with diabetes can develop DR during their lifetime, but the condition is rapidly increasing due to lack of awareness and delayed diagnosis. Once considered a disease of the rich, diabetes has now reached epidemic proportions in both developed and developing countries. At least 366 million people worldwide currently have diabetes and this number will increase due to the world's aging population

Worldwide, the number of people with DR will increase from 126.6 million in 2010

By 2030 this number will reach 191 million and We estimate that unless urgent action is taken, the number of people living with blindness due to diabetic retinopathy (VTDR) will increase from 37.3 million to 56.3 million. Diabetes is the best disease to use deep learning models. There are many experts working on predicting diabetes and the complications caused by diabetes. There are many apps that can help doctors examine diseases and problems, but most have their own advantages and disadvantages. According to the presentation, most Indians suffer from diabetes due to many factors such as lifestyle, nutrition and lack of physical activity. Diabetic retinopathy is a serious problem that affects the eyes of people with diabetes. Damage to the blood vessels in the retina tissue causes the disease. Diabetic retinopathy (DR) is a type of diabetes that causes the blood vessels in the retina to swell and turn into fluid and blood. It is the leading cause of blindness in people ages 20 to 64. Diabetic retinopathy (DR) is the leading cause of blindness in humans worldwide. According to the article, it is claimed that approximately one-third of 285 million people have diabetes Symptoms of diabetic retinopathy are seen worldwide.

EXISTING SYSTEM

Existing methods were trained in addition to the binocular model for various diabetic retinopathy tests and further evaluations were conducted to demonstrate the effectiveness of the binocular model. The final results show that the kappa score of the binocular model in 10% of the reference set was 0.829; this is higher than the current model without integration. Finally, the confusion matrices obtained from models with and without integration were compared and it was proven that the binocular architecture actually improved the dispersion.

PROBLEM STATEMENTS

The problem addressed in this study is the need for a cost-effective and scalable solution for the detection of diabetic retinopathy (DR), a leading cause of blindness in adults with diabetes. Current methods for DR detection are time-consuming, expensive, and require specialized equipment and trained personnel, making them inaccessible in many regions. The objective of this study is to develop and evaluate a deep learning model for automated DR detection from retinal images, which can improve the accessibility and affordability of DR screening and diagnosis.

MOTIVATION

Efforts to develop deep learning models for the diagnosis of diabetic retinopathy are a serious burden that this disease poses on patients, healt hcare systems and society. Early detection and treatment of DR is critical to preventing blindn ess and improving patient outcomes. However, current DR diagnostic methods are expensive a nd impractical in many regions, causing delays in diagnosis and treatment. Using deep learnin g to improve the quality and measurement of D R, this research aims to improve the accessibili ty and capacity of DR screening and diagnosis, while ultimately improving patient outcomes a nd reducing healthcare System.

Scope

The scope of this study is to develop and evaluate a deep learning model for automatic detection of diabetic retinopathy (DR) from retinal images. This work will train and validate a deep learning model using visual information and recording DR weights. Deep learning models use a combination of convolutional neural networks (CNN) and recurrent neural networks (RNN) for feature extraction, sequence modeling, and DR cardinality classification. The performance of the deep learning model will be compared with other state-of-the-art DR methods to evaluate its effectiveness and efficiency. Deep learning models can have significant clinical implications, including early detection and treatment of DR, reducing ophthalmologists' workload, and increasing patient benefit. However, studies may encounter limitations such as low sample size and biased data, which may affect the deep learning model. The scope of this study is limited to the development and evaluation of the deep learning model for automatic DR detection and does not include its implementation and validation in a real clinical setting.

OBJECTIVES

The objectives of this study are:

1. Develop a deep learning model to detect diabetic retinopathy (DR) from retinal images.
2. Evaluate the effectiveness of deep learning models using standard methods (such as regress ion and regression) and performance metrics (such as accuracy, precision, and detail).
3. Compare the performance of deep learning m odels with other DR state detection methods to evaluate their superiority and effectiveness.
4. Describe the importance of deep learning m odels, including early detection and treatment of DR, reducing ophthalmologists' workload, and improving patient outcomes.
5. Identify limitations or issues encountered in t he development and evaluation of deep learning models that may affect the details of the model, such as limited sample size or biased data.
6. Provide recommendations for future research in DR research using deep learning such as mod eling and validation in real clinical settings.

Proposed System

The system proposed in this study is a deep lea rning model for automatic detection of diabetic retinopathy (DR) from retina images. The mo del uses a combination of convolutional neural networks (CNN) and recurrent neural network s (RNN) for video extraction, modeling, and w eighted DR classification. The deep learning m odel will be trained and validated using visual information and DR weights. The performance of the model will be evaluated using standard methods such as back expansion and regressio n, as well as performance measures such as ac curacy, desirability and specificity. The propos ed system is intended to provide the most effec tive and efficient solution for DR detection, th us providing easy and affordable access to DR screening and diagnosis. The impact of this pr ocess includes early detection and treatment of DR, reduced physician workload, and improv ed patient outcomes. The limitations and probl ems of the proposed method, such as low samp le size and biased data, will also be evaluated i n the study.

Advantages

1. Accuracy: Deep learning models provide high accuracy in diagnosing diabetic retinopathy.
2. Effective: Using deep learning techniques ca n speed up the process of diagnosing diabetes a nd retinopathy.
3. Cost-
effective: Using deep learning to diagnose diab etic retinopathy is a cost-
effective alternative to screening. 4. Scalability: Deep learning models can be easily extended t o analyze large data sets and adapt to other dise ases and conditions.
5. Access: Automated diabetic retinopathy dete ction using deep learning can improve access to screening and diagnosis in underserved areas.
6. Sensitivity: Deep learning models are sensiti ve in detecting early signs of diabetic retinopat hy. 7. Reduce work: Using deep learning to dia gnose diabetic retinopathy can reduce the work of ophthalmologists and doctors.
8. Purpose: In-
depth study provides an objective and consisten t approach to the diagnosis and diagnosis of dia betic retinopathy.
9. Early intervention: Early detection of diabeti c retinopathy using deep learning can lead to ea rly intervention and treatment. 10. Improve pati ent outcomes: Using deep learning to diagnose diabetic retinopathy can improve patient outco mes and quality of life.

Disadvantages

1. Lack of transparency: Deep learning models can be complex and opaque, making decisions difficult to understand.
2. Dependency on data: Deep learning models r ely heavily on good data, which may not be ava ilable.
3. Overfitting: Deep learning models can overfi t training data, leading to poor performance and underperformance on new data. 4. Hardware re quirements: Deep learning models require a lot of computing power, making it difficult to depl oy on low-end devices.

5. Disadvantages of hacking attacks: Deep lear ning models can be vulnerable to attacks, subtle changes in input data result in the wrong
product. 6. Limitations: Deep learning models c annot provide good insight into the biological p rocesses or mechanisms of diabetic retinopathy
. 7. Ethical issues: Using deep learning to diagn ose diabetic retinopathy raises ethical questions regarding privacy, private data, and bias.

8. Maintenance and updates: Deep learning models require maintenance and updates to ensure performance and accuracy. 9. Limited availabil ity: Availability of deep learning models and ca pabilities may be limited in certain regions or m edical systems. 10. Regulatory and Legal Decis ions: Use of Diabetic Retinopathy Research may be subject to legal and regulatory consider ations, particularly with respect to patient safet y and liability.

Research articles

1. Gargeya R, Leng T. Automatic identificatio n of diabetic retinopathy using deep learning. Ophthalmology. 2017; 124(7): 962-
969. doi:10.1016/j.ophtha.2017.02.008.
2. Gulshan V, Peng L, Coram M, et al. Develo pment and validation of a deep learning algorit hm to detect diabetic retinopathy in retinal fun dus images. Jama. 2016; 316(22): 2402-2410. doi:10.1001/jama.2016.17216.
3. Abromoff MD, Lou Y, Erginay A, et al. Imp roved automatic detection of diabetic retinopat hy in population data through integrated deep l earning. Investing in vision research in ophthal mology.
2016;57(13):5200-5206. doi: 10.1167/iovs.16-
19964. 4. Ding DSW, Zhang CY, Lin G, et al. Development and validation of an in-
depth study of diabetic retinopathy and eye dis eases using retinal images from a variety of dia betic patients. Jama. 2017; 318(22): 2211-
2223. doi:10.1001/jama.2017.18152.
5. Rajalakshmi R, Subashini R, Anjana RM, M ohan V. Automatic detection of diabetic retino pathy using artificial intelligence in smartphon e-
based fundus photography. Eyes. 2018; 32(6):
1138-1144.
doi: 10.1038/s41433-018-0078-
y. 6. Li Z, Kiel S, Liu C, et al. Automated grad ing system to detect complications of diabetic retinopathy based on

color image. Diabetes. 2018; 41(12): 2509-2516. doi: 10.2337/dc18-
0685. 7. Bellemo V, Lim ZW, Lim G, et al. Ar tificial intelligence using deep learning to scre en for use and visual interventions of diabetic r etinopathy in Africa: a clinical study. Lancet D igital Health. 2019;1(1):e35-
e44. doi: 10.1016/S2589-7500(19)30010-9.
8. Saba L, Biswas M, Kupili V, et al. Automat ed detection of diabetic retinopathy using optic al coherence tomography angiography: A study. Br J Ophthalmol. 2018; 102(11): 1564-
1569.
doi: 10.1136/bjothamol-2017-311617.
9. Wang Y, Ji Y, Chen S, et al. Diagnostic reti nopathy based on deep convolutional neural ne twork transfer learning. Computational Biome dicine. 2018; 102: 293-301.
doi:10.1016/j.compbiomed.2018.09.004.
10. van der Heijden AA, Abramoff MD, Verbr aak F, et al. Validation of automated screening for referral for diabetic retinopathy using the I Dx-
DR tool in the Hoorn Diabetes Care System. D iabetes
Treatment. 2018; 41(12): 2535-
2542. doi:10.2337/dc18-1119.

System Requirements Specification Functional Requirements:

The functional requirements of the deep learni ng system for diagnosis of diabetic retinopathy include:
1 . Image capture: The system must be able to capture good quality images for analysis.
2. Image pre-
processing: The system must be able to pre- process images to remove noise, adjust contras t and improve focus.
3. Diabetic retinopathy diagnosis: The system should be able to detect diabetic retinopathy in retinal images using deep learning. 4. Classifi cation: The system should be able to classify d iabetic retinopathy into different stages accordi ng to the severity of the disease.
5. User Interface: The system should be intuiti ve and easy to use for doctors to interact with t he system. 6. Patient database: The system mu st be able to manage the patient database and s tore patient information and physical images. 7
. Reporting: The system should produce infor mation about the results of each diabetic retino pathy diagnosis and classification.
8. Integration with Electronic Health Records (EHR): Systems must be able to integrate with EHRs to facilitate patient care and managemen t.
9. Security: The system must have good securi ty to protect the patient's information and com ply with relevant regulations.

10. Maintenance and Updates: The system should be easy to maintain and update to ensure ef ficiency and accuracy.

NON-FUNCTIONAL REQUIREMENTS:

1. Performance: The system must be fast and e ffective to ensure timely diagnosis and treatme nt.
2. Accuracy: The system should have a high ac curacy in diagnosing diabetic retinopathy, redu cing false positives and negative results.
3. Robustness: The system must be able to han dle changes in the retinal image based on age, race, and image quality. 4. Scalability: The sys tem must be able to manage multiple patient re cords and adapt to other diseases and condition s.
5. Usability: The system should be easy to use and require minimal training on the operating s ystem. 6. Reliability: To ensure uninterrupted patient care, the system must be reliable and ac cessible 24/7. 7. Interoperability: Systems mus t be able to integrate with other healthcare syst ems and technologies to support patient care a nd management.
8. Data privacy and security: The system must have data privacy and security measures to pro tect patient information and comply with relev ant regulations.
9. Maintenance and support: The system shoul d be easy to maintain and provide adequate su pport to users. 10. Accessibility: The system s hould be accessible to all patients regardless of location, language or disability.

Hardware Requirements

™ System: Pentium IV 2.4 GHz/intel i3/i4.
> Hard drive: 40 GB. Monitor: 15 VGA Color. Memory: At least 512 Mb
3.2 Required software
Operating system: Windows XP/ Windows10
™ Software package: Tensorflow, OpenCv
™ Coding language: Python.

Software Requirements

Operating system:Windows XP/ Windows10 Software Packages : Tensorflow , OpenCv Coding Language :Python.

SYSTEM DESIGN

DesignOverview

Design overview explains the architecture that would be used for developing a software product. It is an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

System Architecture

CNN is a type of DNN that includes many hid den operations such as convolutional operation s, RELU layers, pooling layers and all normali zation operations. CNN reported.
Weighing in layers reduces memory usage and improves network performance. The main feat ures of CNNs are 3D volume, local connectivit y, and mutual integration of neurons. The conv olutional process creates feature maps by conv olving different subregions of the input image with the learning kernel. A non-linear activation method is then used by the Re Lu layer to improve integration while the error is low. In the pooling layer, select an area of the image/feature map and select the pixel wit h the maximum or average value as a represent ative.

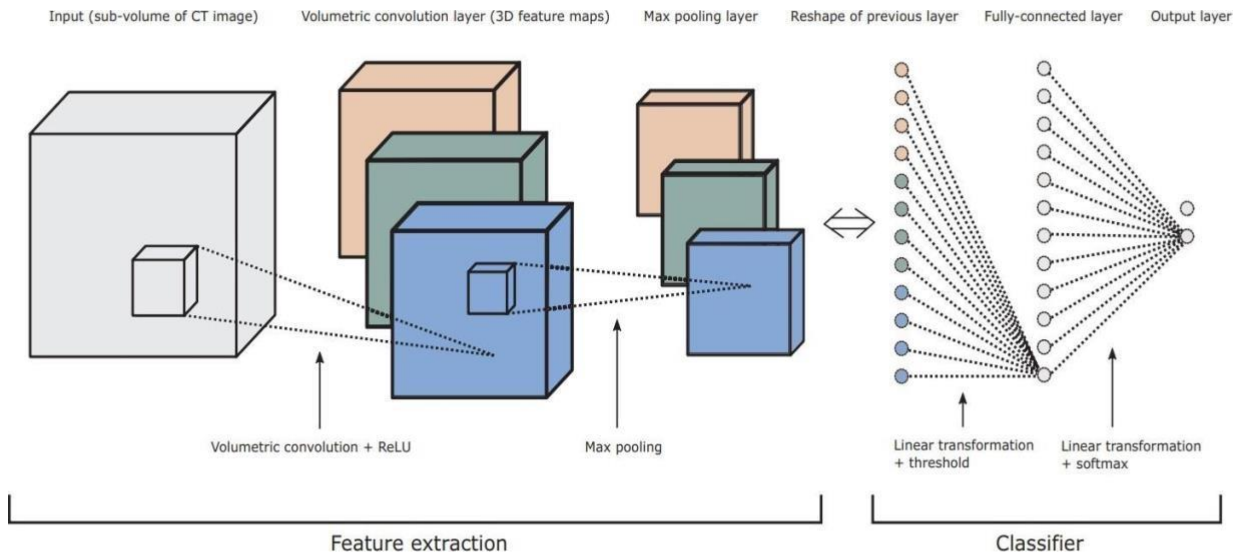


Fig:Deep-Convolutional Neural Network Architecture.

This causes the sample size to decrease. In so me cases, fully connected layers (FC) are used with layers in the output stage. Convolution an d pooling layers are often used together in CN N architecture. Pooling layers generally do tw o types of jobs. Maximum pooling
means pooling. In point pooling, the average is

calculated at the community feature scores, w hile in maximum pooling, it is calculated at the largest feature point. Average pooling reduces the error caused by limiting the size of the ens emble and retaining background information. Maximum pooling reduces the prediction error of the convolution process caused by the mea n difference, thus keeping the data more beauti ful.

System architecture for diagnosing diabetic ret inopathy using deep learning can include many aspects, including:

1. Image acquisition: Retinal cameras or devic es are actually used to acquire good objects of the patient's retina image. .
2. Preprocessing: Use noise reduction, contrast
, image enhancement, and other preprocessing techniques for retina images to improve image quality.
3. Feature extraction: Use deep learning algorit hms to extract relevant features from images b efore processing.
4. Training and validation: In troducing deep learning models on large datase ts of retinal images containing diabetic retinop athy markers. The model is validated on a sepa rate dataset to evaluate its performance.
5. Evaluation and evaluation: Evaluate the training model of the new retinal image to evaluat e its accuracy and performance. 6. User Interfa ce: A user-
friendly interface has been designed so that do ctors can interact with the system and access p atient information. 7. Patient Database: Stores patient information to store patient information and retinal images for future reference.
8. Description: A guideline was developed to p rovide information on diabetic retinopathy scre ening and classification results for each patient
9. Integration with EHR:The system integrates with electronic health records (EHR)
to streamline care and patient management.
10. Maintenance and Updates: Designed to be easy to maintain and update to ensure the syste m operates efficiently and on time.

Convolutional Neural Network:

CNN is a type of DNN with many hidden laye rs such as layers, RELU layers, pooling layers and fully connected normalization layers. CN Ns reduce memory usage and improve networ k performance by weighting layers. The main f eatures of CNNs are 3D volume, local connect ivity, and mutual integration of neurons. Featu re maps are generated by convolutional layers by convolving different subregions of the input image with the learning kernel. A non-
linear activation method is then used by the Re Lu layer to improve integration while the error is low. In the pooling layer, select an area of the image/feature map and select the pixel wit h the maximum or average value as the represe ntative pixel, thus reducing the 2x2 or 3x3 grid to a scalar value. This causes the sample size t o decrease. In some cases, fully connected layers (FC) are used with other layers in the output stage.

CNN has several layers:

Convolution layer: Applies a filter that evaluat es the entire image (two to three pixels at a tim e), creating a custom map that estimates the cl ass probability of each feature.

Pooling layer (subsampling):
a convolutional layer that reduces the data valu e and checks the most important data for each feature (the process of convolution layer and p ooling layer respectively repeats
many times).

The combination is fully understood: Flattenin g the output of the previous layer transforms th em into a vector that can be used as the input o f the next layer. Fully anchored method: Apply ing weight to ideas resulting from a particular analysis to estimate facts.

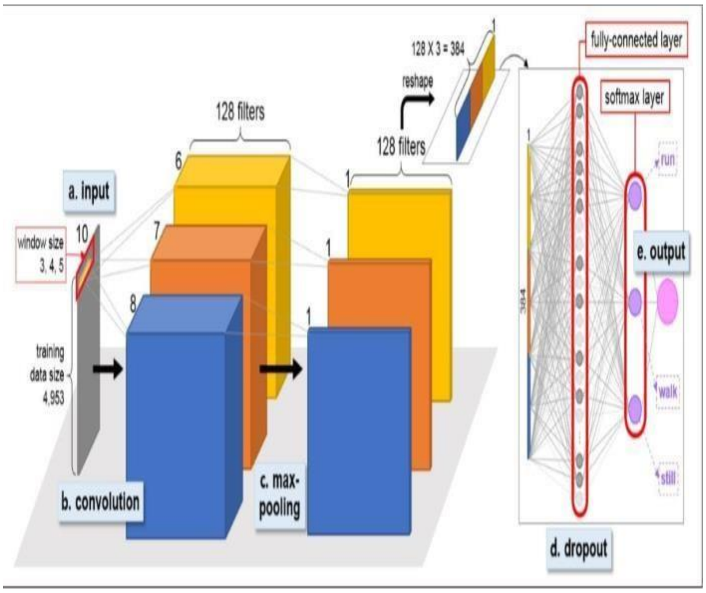
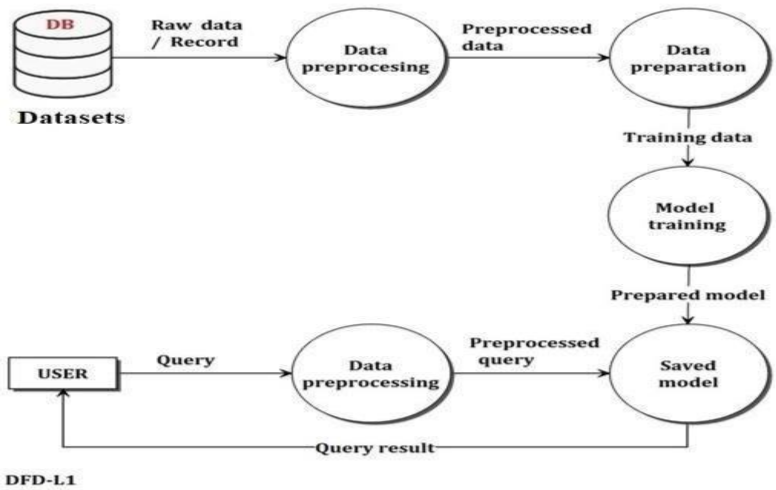


Fig: Convolutional Neural Network General Architecture

Data flow diagrams

A dataflow outline is a tool for referring to knowledge progression from one module to the next module as shown in Fig 4.3 This graph gives the data of each module's info and yield. The map has no power flow and there are no circles at the same time.

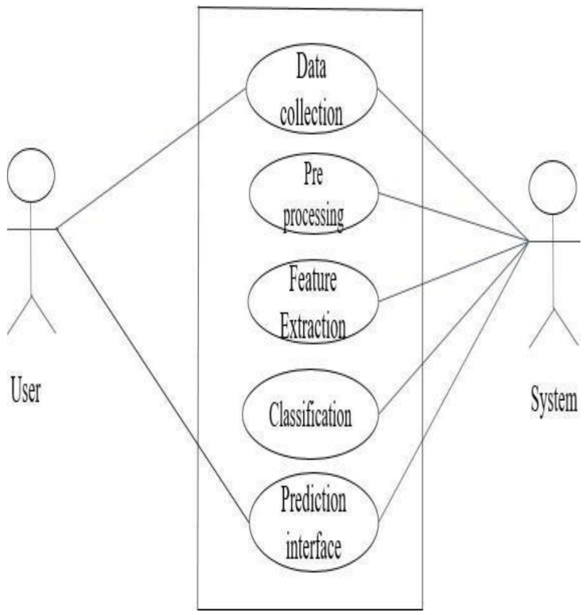


Data Flow Diagram

Use CaseDiagrams

Use case diagram is the boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which are the specific roles played by the actors within and around the system.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

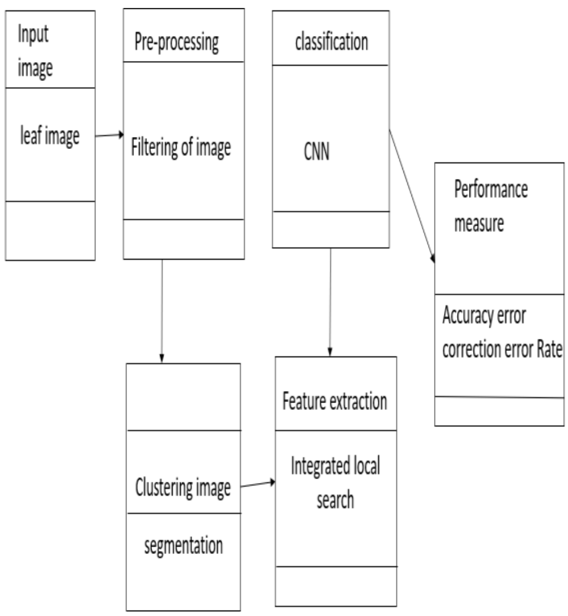


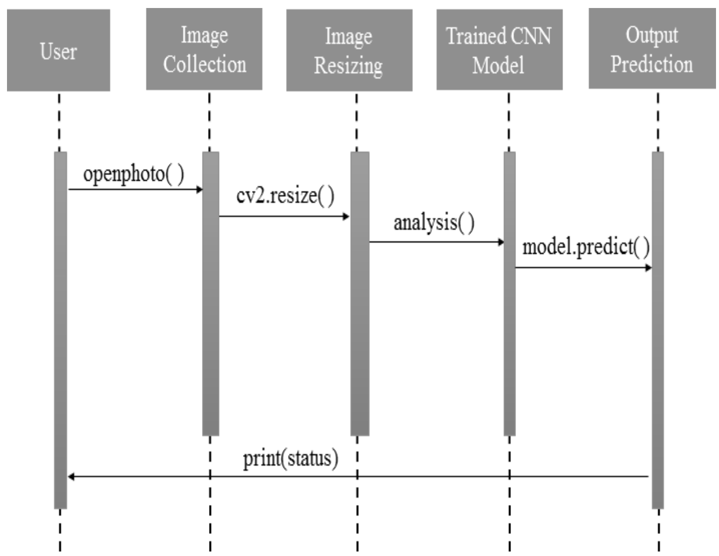
Class Diagrams

Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them as shown in the Fig

Sequence Diagrams

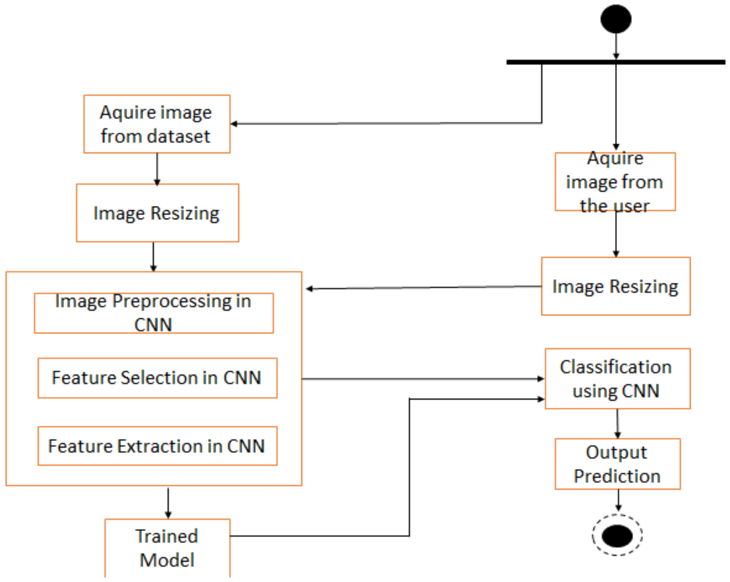
A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place as shown in Fig

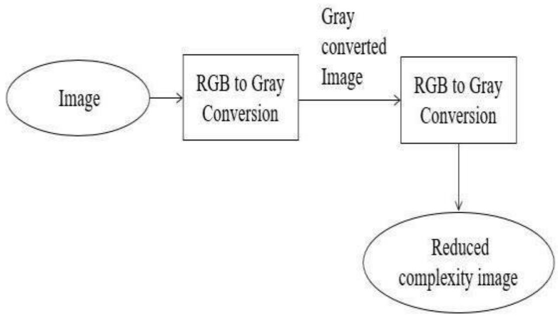




Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by- step workflows of components in a system. An activity diagram shows the overall flow of control.





Here is an example of a data flow diagram for pre-processing of retinal images for diabetic retinopathy detection using deep learning:

[Input]: Raw retinal images

[Process 1]: Image cropping - The retinal images are cropped to remove any areas of the image that are not relevant to diabetic retinopathy detection.

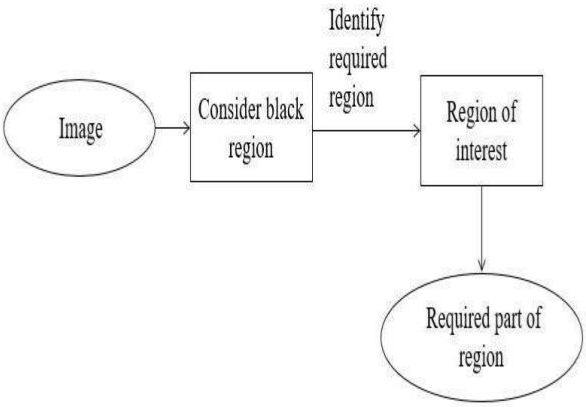
[Process 2]: Image resizing - The cropped images are resized to a standard size to ensure consistency in feature extraction.

[Process 3]: Image normalization - The resized images are normalized to adjust the contrast and brightness, remove any noise, and enhance relevant features.

[Output]: Preprocessed retinal images ready for feature extraction and deep learning algorithms.

[Feedback loop]: The output of the pre- processing stage is fed back into the deep learning model for feature extraction and diabetic retinopathy detection. If the detection results are not satisfactory, the pre-processing stage may need to be adjusted and repeated to improve image quality and accuracy of detection.

Data Flow Diagram for Identification



Here is an example of a data flow diagram for identification of diabetic retinopathy using deep learning:

[Input]: Preprocessed retinal images

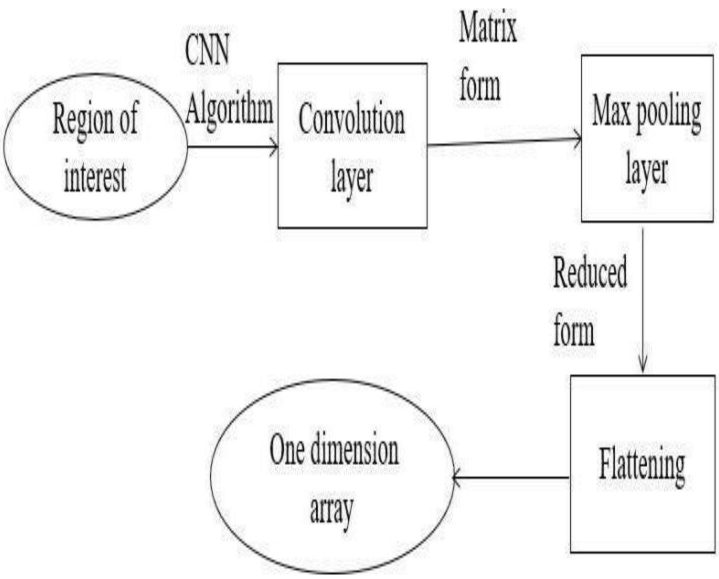
[Process 1]: Feature extraction - Deep learning algorithms are used to extract relevant features from the preprocessed retinal images.

[Process 2]: Diabetic retinopathy detection - The extracted features are used to classify the retinal images into categories of diabetic retinopathy severity.

[Output]: Detection results - The system generates detection results indicating the severity of diabetic retinopathy in the retinal images.

[Feedback loop]: The detection results are fed back into the system to evaluate the performance of the deep learning algorithms. If the accuracy of detection is not satisfactory, the system may need to be adjusted and retrained to improve its performance.

Data Flow Diagram for Feature Extraction



Here is an example of a data flow diagram for feature extraction in diabetic retinopathy detection using deep learning:

[Input]: Preprocessed retinal images

[Process 1]: Convolutional neural network (CNN) - A CNN is used to analyze the preprocessed images and extract

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

relevant features.

[Process 2]: Feature selection - The CNN outputs a set of features, and a feature selection algorithm is used to identify the most important features for diabetic retinopathy detection.

[Process 3]: Feature normalization - The selected features are normalized to ensure consistency in classification.

[Output]: Extracted features - The system generates a set of extracted features that are used in the diabetic retinopathy detection process.

[Feedback loop]: The extracted features are fed back into the system to evaluate the performance of the deep learning algorithms. If the accuracy of detection is not satisfactory, the system may need to be adjusted and retrained to improve its performance.

Data Flow Diagram for Classification and Detection

Here is an example of a data flow diagram for diabetic retinopathy classification and detection using deep learning:

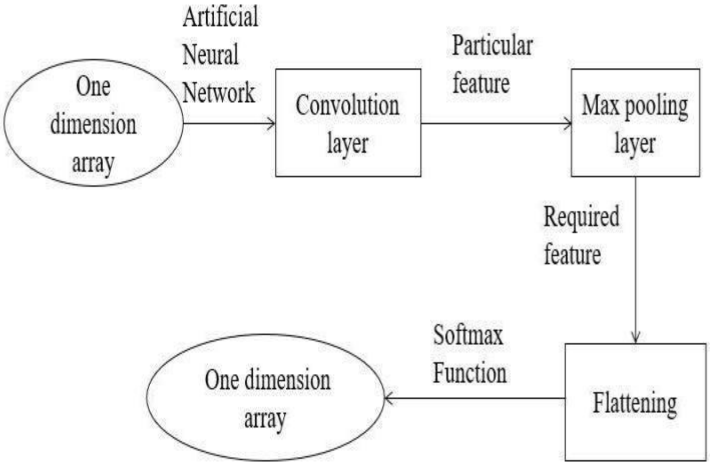
[Input]: Extracted features

[Process 1]: Classification - The extracted features are used to classify the retinal images into categories of diabetic retinopathy severity using a deep learning model such as a support vector machine (SVM) or a random forest classifier.

[Process 2]: Detection - The classified images are analyzed to detect signs of diabetic retinopathy, including microaneurysms, hemorrhages, and exudates.

[Output]: Detection results - The system generates detection results indicating the severity of diabetic retinopathy in the retinal images and the location of any abnormalities.

[Feedback loop]: The detection results are fed back into the system to evaluate the performance of the deep learning algorithms. If the accuracy of detection is not satisfactory, the system may need to be adjusted and retrained to improve its performance.



IMPLEMENTATION

Implementation is the process of turning a new design into action. This is an important step in the implementation of the new method. Theref ore, it needs to be carefully planned and mana ged. System application after successful devel opment
Working steps
Front-end development using Python Flask: Modern computer applications are users It is a good product. User

interaction is not limited to console-
based I/O. They have more ergonomic graphic al user interfaces (GUIs) thanks to high-
speed processors and powerful graphics hardw are. These applications receive input through mouse clicks and allow users to choose from al ternatives using radio buttons, drop-
down lists, and other GUI elements.

Flask Programming:

Flask is a standard GUI library for Python. Pyt hon provides a fast and easy way to create GU I applications when used with Flask. Flask pro vides strong object orientation for Tk GUI tool
s. The bottle has many advantages. Since it is c ross-
platform, the same code can run on Windows, macOS and Linux. Content visualization is do ne using native workflows, so applications buil t with Flask look just like the platform they ru n on

.

Implementation issues

The software development process is about tra nslating the design into source code. The main purpose of the application is to write the sourc e code and internal documentation so that the c ode can be easily identified in terms of its com pliance with the instructions, thus making it ea sier to debug, test and modify. This goal can b e achieved by making the source code as clear and simple as possible. Simplicity, clarity and elegance are the hallmarks of good work, and t hese characteristics are achieved in every prog ramming module.
The targets to be achieved are as follows. Minimize memory requirements.
Maximize the readability of your output. Increase the readability of your text.
Minimize the number of instructions. Shorter development time

Module specification:

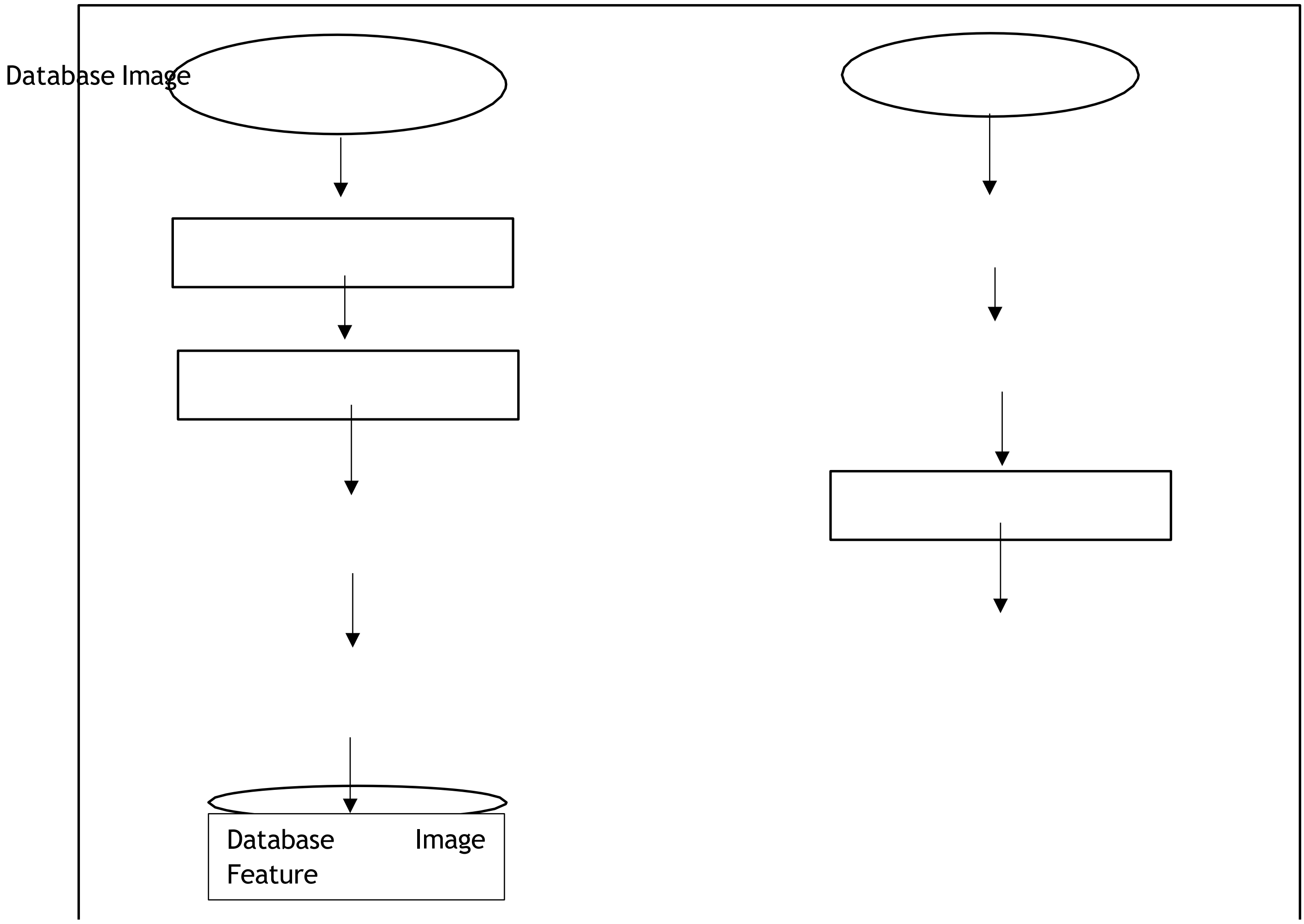
Module specification improves the design by b reaking the system into modules and solving th em as an independent method. Doing this will reduce complexity and allow the module to tes t itself. The number of models in our model is three: preprocessing, recognition, feature extra ction and detection. Therefore, each stage repr esents the work provided by the planning proc ess. In the first case, median filtering is used to eliminate the noise level.

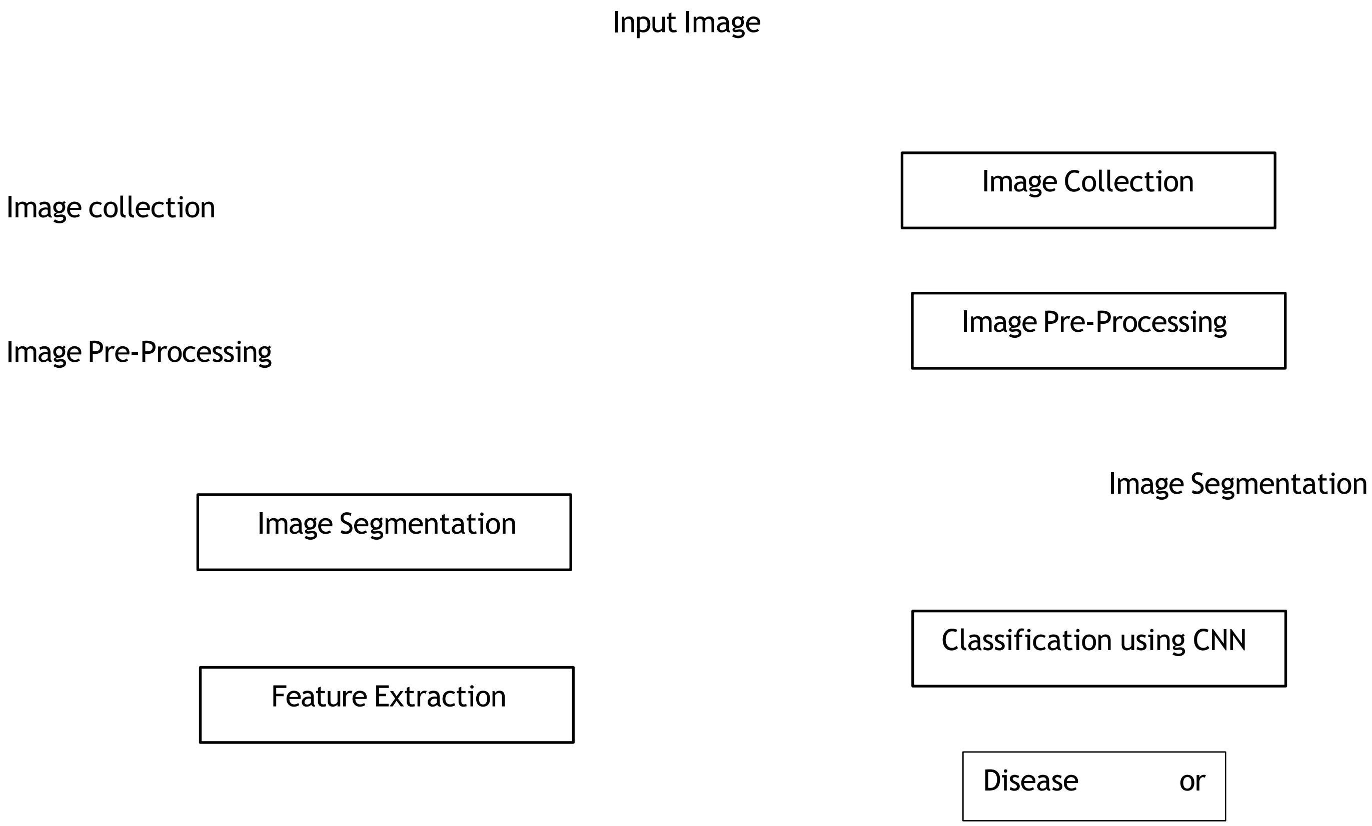
Data flow Diagram of Training and Testing Phase

The System design mainly consists of

1. Image Collection
2. Image Preprocessing
3. Image Segmentation
4. Feature Extraction
5. Training 6. Classification

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930




Input Image
Image collection
Image Collection
Image Pre-Processing
Image Pre-Processing
Image Segmentation
Image Segmentation
Classification using CNN
Feature Extraction
Disease or

1. Image collection

The information we used in this project is publi cly available online

2. Image Preprocessing

The purpose of preprocessing is to improve ima ge data, reduce unnecessary distortion, and imp rove some image features that are important for further image processing. There are three main features of image preprocessing: a) Grayscale c onversion b) noise removal c) Image enhancem ent

a) Grayscale conversion: Grayscale images hav e only brightness information. Each pixel value in a grayscale image corresponds to a value or a mount of light. Brightness levels can be disting uished from grayscale images. Grayscale image s measure light intensity only. The brightness o f an 8-
bit image ranges from 0 to 255; where "0" repre sents black and "255" represents white. In grays cale conversion, color images are converted to grayscale images for display. Creating grayscal e images is easier and faster than color images. For grayscale images, all image processing met hods are used.

b) Noise cancellation: The purpose of noise can cellation is to capture and eliminate unwanted n oise in digital images. The challenge is to deter mine which features of the image are real and w hich are due to noise. Noise is a random change in pixel value.

We use median filter to eliminate unwanted noi se. The median filter is a nonlinear filter that lea ves the edges unchanged. The median filter is i mplemented with a sliding window of variable l ength. Each instance value is sorted by size; Th e most important value is the average value of t he sample in the window that is the filter.

c) Image enhancement: The purpose of image e nhancement is to ensure that the image displays the features of interest. A good comparison has been used here to get better results.

3. Image Segmentation There are many types of image segmentation such as stitching, threshol ding, neural network-based, and edge-
based. In this application, we use a clustering al gorithm called mean-
shift clustering for image segmentation. This al gorithm uses the sliding window method to find the location of the largest density. This algorith m uses multiple sliding windows to collect max imum space. Mean change group algorithm Thi s algorithm is used only to determine the high p oint.

Feature extraction:

The image has many features, especially color, texture and shape. Here we consider three thing s: color histogram, color-
like texture, shape, and texture.

5. Training Training data is generated from ima ges of known cancer cells. The classifier is train ed from the generated training data. The test fil e is placed in a temporary folder. Predict the res ults of the test data, draw separate images, and a
dd custom parameters to the test data to make th
e image processing model more accurate

6. Divide into two sets of binary classifiers usin g hyperplanes (also called hyperplanes). Decisi on boundary The structure of the group is called convolutional neural networks. Some of these problems are familiar models, such as texture cl assification using CNNs. Nonlinear mapping of input data to output data can achieve good clas sification in the high-
speed domain of CNNs. CNN is widening the g ap between different groups. Use different kern els to split the clusters. CNN is a binary classifi er that determines the hyperplane that divides t wo classes. The hyperplane and boundary betw een two classes are maximized. When determin ing the hyperplane, the structures closest to the edges, called support vectors, are selected.

Explanation of CNN Algorithm Convolutional Neural Network:

Convolutional neural network is a special type of feedforward artificial neural network in whic h the connections between layers are created by the visual cortex. A convolutional neural netw ork (CNN) is a type of deep neural network use d to analyze visual images. Image and video rec ognition, image classification, natural language processing, etc. It has applications in different f ields. Convolution is the first layer that extracts features from the input image. Convolution pre serves the relationship between pixels by learni ng image features using least squares of the inp ut data. It is a mathematical operation that requires two ideas such as matrix image and filter or kernel. Each input image will be passed throug h a convolutional process with filters (kernels) t o create an output map. This is the way CNN w orks.

Essentially, the convolutional neural network h as 4 layers; layer by layer, ReLU layer, layer by layer and fully connected layer.

Convolutional Layer

After the computer reads the image in the form of pixels in the convolution, we use the convolu tional layer and then we obtain a small-

sized part of the image with the help of the conv olutional layer. These images or patches are cal led features or filters. By sending these rough m atches to the same area in two images, the conv olution technique can show the similarity better than whole image comparison. These filters are compared with the new input image and if it m atches, the image is classified correctly. Here fe atures and images are tracked, then each image pixel is divided into corresponding pixels, the p ixels are counted and divided by all pixels in th e feature. We create a chart and place the filter values

in the relevant area. Likewise, we will move the

feature anywhere in the image and see how the feature matches the area. Finally, we will get a matrix as output.

ReLU Layer

ReLU layer is nothing but changing the lines, in this layer we remove all negative values

from the filter image and replace it with zero. T his is to prevent the sum of the values

from falling to zero. This is a variable that initia lizes the node only when the input value is great er than a number and the input value is less than zero, the output will be zero, and then deletes a ll values of the matrix.

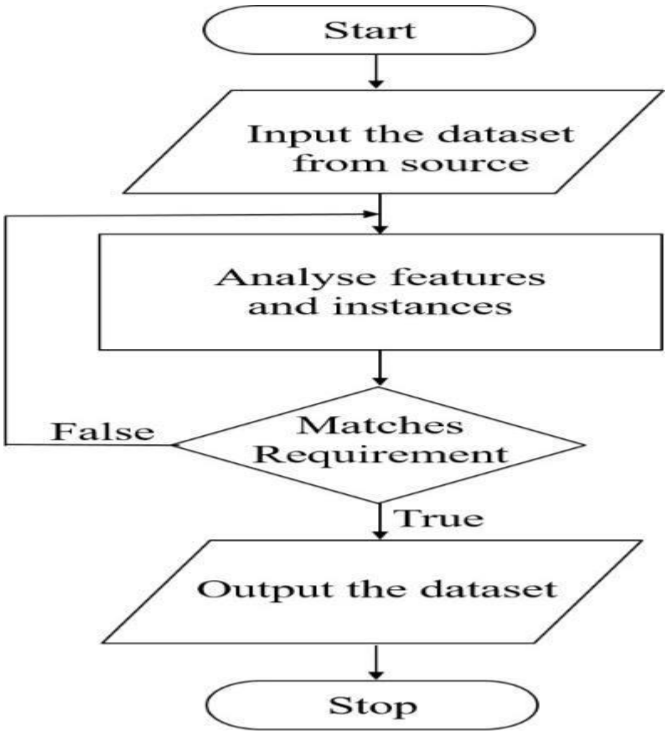
Pooling Layer

In this layer we reduce or reduce the size of th e image. First we select the size of the viewpor t, then the desired step, and then we move our viewport above the view filter. Then take the h ighest value from each viewpoint. This will me rge the layers and reduce the size of the image and matrix. Use the reduced size matrix as input for the full link operation.

Full connection method

After passing layer by layer, ReLU layer and l ayer by layer, we need to pack all the layers. T he full link layer is used to classify the input i mage. Layers should be repeated if necessary, unless you end up with a 2x2 matrix. And final ly use the link process where the actual deploy ment happens.

The flowchart for Data Acquisition



Flowchart for data acquisition.

The flowchart for collecting data is as depicted in the figure 4.3. The data set is collected from a source and a complete analysis is carried out. The image is selected to be used for training/testing purposes only if it matches our requirements and is not repeated.

Flowchart for Pre-Processing the Data Set

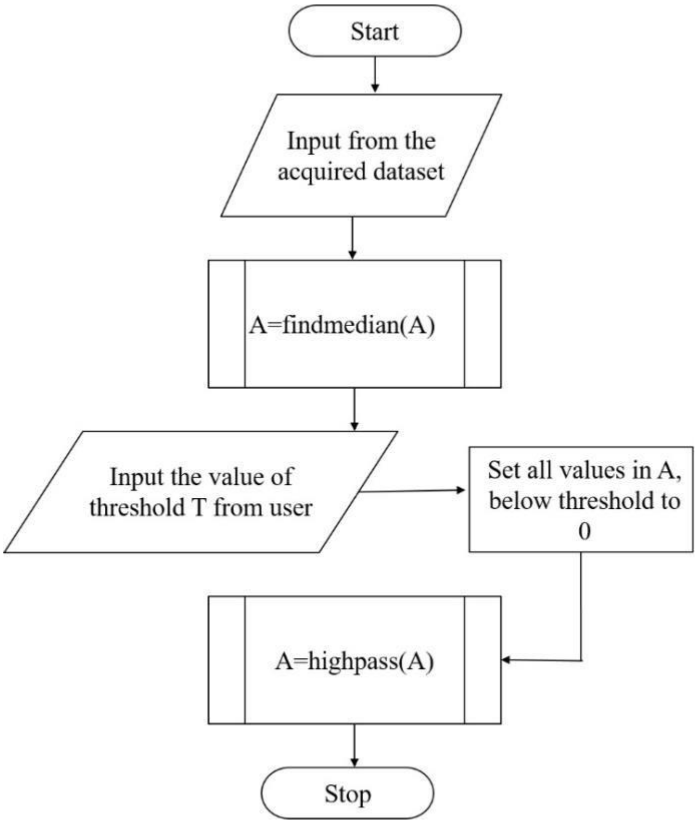


Figure:Flowchart for the preprocessing module

The figure 4.4 shows the flowchart for the pre- processing of the images received from the output of the previous step. This involves converting the image from the RGB format to greyscale to ease processing, the use of an averaging filter to filter out the noise, global basic thresholding to remove the background and consider only the image and a high- pass filter to sharpen the image by amplifying the finer details.

Conversion from RGB to Greyscale

The first step in pre-processing is converting the image from RGB to Greyscale. It can be obtained by applying the below formula to the RGB image. The figure 4.5 depicts the Conversion from RGB to grayscale.

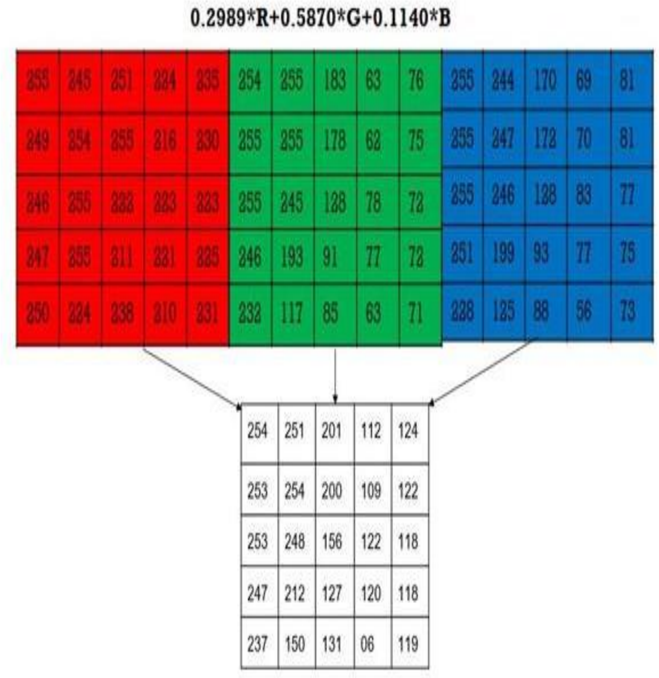


Figure Converting from RGB to grayscale

Advantages of converting RGB color space to grayscale
ï To store the monochrome image of RGB colo r image we need 8 * 3 = 24 elements (8 eleme nts) per color component).
Only 8 objects are required to store one pixel o f an image. Therefore, we need 33% less mem ory to store grayscale images compared to stor ing RGB images.
· Grayscale images are easier for many tasks, s uch as many morphological tasks and image se gmentation problems, it is better to create singl e layer images (gray images) than to make Thr ee layer images (RGB color image).
ï¤ When we process images in a layer, it is eas ier to distinguish images from each other
â ¢ Denoising

Noise removal algorithms remove or reduce no ise in the standard image of layers. Noise redu ction algorithms reduce or eliminate the appear ance of noise by moving the entire image out o f the immediate area of contrast. Noise removal is the second step of i mage processing. The grayscale image obtaine d in the previous step is provided as input here.

Here we use median filter, which is a noise re moval technique.
➞ Median filter
A median filter is a nonlinear digital filter ofte n used to remove noise in images or symbols.
< br> 0 is added here to the edges and corners of the matrix, representing the grayscale image
. Then for each 3*3 matrix, arrange the points i n ascending order and then find the mean/medi an of the 9 points and record the average for a particular pixel position. Figure 4.6 shows nois e filtering using the median filter.

Figure Noise filtering using median filter Basic general threshold

Threshold is a type of segmentation of images. We change the pixels of the image to make it easier to analyze the image. A(i,j) greater than or equal to the threshold T is preserved. Other wise, replace the value with 0.

Here the T value can be adjusted in the front e nd to accommodate different image sizes. We use trial and error here to find the threshold tha t works best for us.

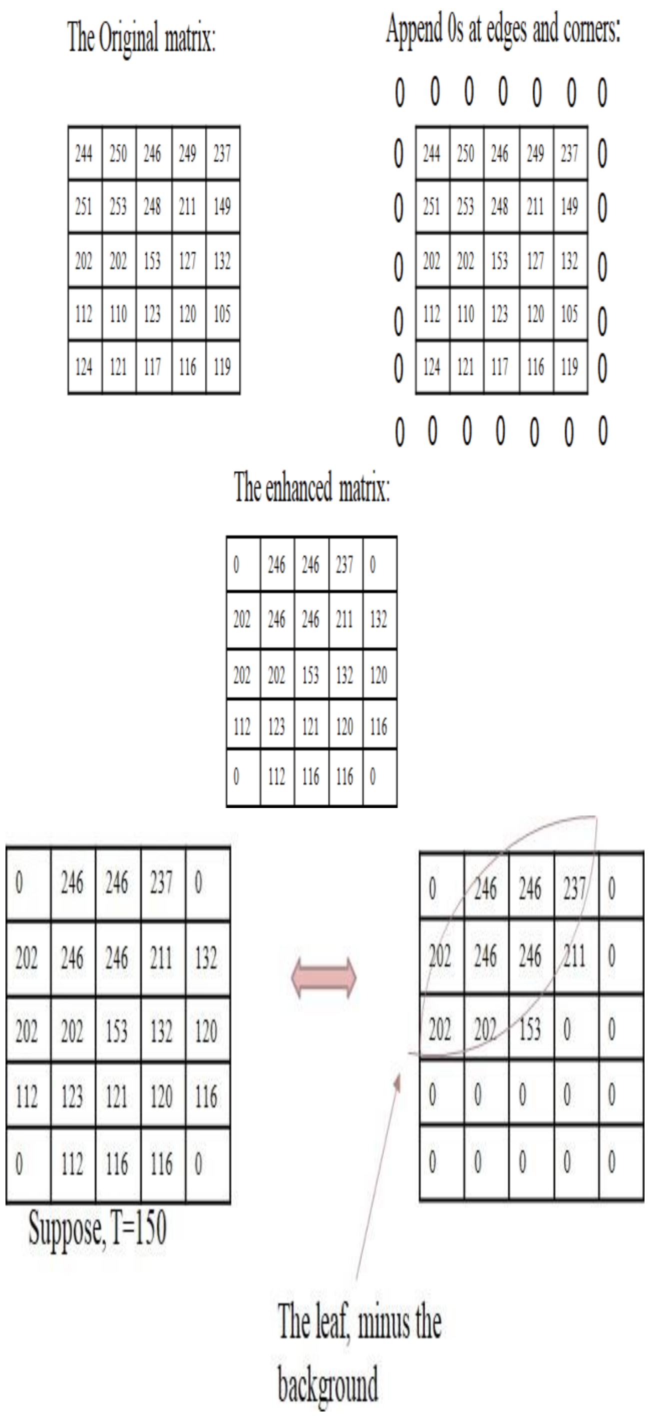


Figure 4.7 Thresholding using Basic global Thresholding

Image Sharpening

Image sharpening refers to any enhancement technique that highlights edges and fine details in an image, Increasing yields a more sharpened image.

High-Pass Filtering

A high-pass filter can be used to make an image appear sharper. These filters emphasize fine details in the image. Here the output from the thresholding is given as input. Here, we are making use of a filter, first we append the nearest values to pixels at the boundary pixels. The figure 4.8 depicts Image Sharpening using High-Pass Filter.