1.INTRODUCTION

Due to their safety and cost-effectiveness in acquiring, retinal fundus images are widely used by both ophthalmologists and computer-aided diagnosis systems for the clinical screening and diagnosis of ocular diseases However, fundus images tend to experience large variations in quality.Recently, general image enhancement methods have achieved state-of-the-art performances, especially with the development of deep learning techniques. However, different from general images, retinal fundus images areacquired through a special ophthalmoscope imaging process to capture anatomical retinal structure for clinical diagnosis, which introduces various additional challenges. First, the retina cannot be illuminated internally; both the incident and reflected imaging beams have to traverse the pupil. Moreover, the spherical geometry of the eye creates significant inter-reflection, resulting in shading artifacts.

A screening study of 5,575 patients found that about 12% of fundus images are not of adequate quality to be readable by ophthalmologists. In some cases, when the degradation is caused by the images being obtained through internal cataractous turbid media, enhancement methods, such as , can be used to restore ‘high quality’. Then, the corrected images can be used to support the observation of other diseases (e.g., age-related maculopathy, diabetic retinopathy, and glaucoma). However, in addition to this pathogenic degradation, in real applications, external interference factors caused by handcrafted imaging equipment and poor environmental conditions are also common. For instance, images are often taken under dd ifferent lighting environments, using various cameras, and by distinct operators with varying levels of experience. Common examples of low-quality factors in retinal fundus images thus include uneven illumination, image blurring, and artifacts, which not only prevent reliable diagnosis by ophthalmologists, but also affect the performance of automated image analyzing system.A clinically oriented fundus enhancement network (cofe-Net) is proposed to suppress global degradation factors, while simultaneously preserving anatomical retinal structures and pathological characteristics for clinical observation and analysis.

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The method utilizes the Inception-v3 model, which is a pre-trained classifier. These algorithms typically constrain the optimal solution using a regularization scheme to solve the non-convex problem. This may incur a heavy computational cost, limiting their applicability in clinical settings..

2.2 PROPOSED SYSTEM

The quality evaluation systems can process based on a convolutional neural network (CNN) to determine the quality fundus image. Evaluation of the quality image based on CNN is trained to determine the good and bad quality of fundus images. In order to improve the accuracy, data augmentation was performed by random horizontal and vertical translation with the interval The optic disc is a critical feature in the retinal fundus image for analysis of glaucoma and orientation of the overall fundus image.

METHODOLOGY

Step1: Collect and store the Data.

Step2: Data pre processing

Step3: Extract the features from dataset

Step4: Make model to learn from the different input features.

Step5: Predict the results

2.3 FEASIBILITY STUDY

A credibility contemplate expects to fair-mindedly and soundly uncover the qualities and inadequacies of a present business or proposed meander, openings and threats present in nature, the benefits required to bring through, and in the long run the prospects for advance. In its most clear terms, the two criteria to judge believability are incurred significant injury required and motivator to the fulfilled.

An inside and out arranged feasibility ponder should give a recorded establishment of the business or wander, a delineation of the thing or organization,

accounting explanations, purposes of enthusiasm of the operations and organization, publicizing examination and game plans, budgetary data, authentic necessities and cost duties. All things considered, plausibility looks at go before specific change and wander utilization. There are three sorts of attainability

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

2.3.1 ECONOMICAL FEASIBILITY

The electronic structure manages the present existing system's data stream and technique absolutely and should make each one of the reports of the manual structure other than a substantial gathering of the other organization reports. It should be filled in as an electronic application with specific web server and database server. Advance a segment of the associated trades happen in different ranges. Open source programming like TOMCAT, JAVA, MySQL and Linux is used to restrict the cost for the Customer. No extraordinary wander need to manage the instrument.

2.3.2 TECHNICAL FEASIBILITY

Surveying the particular probability is the trickiest bit of a believability consider. This is in light of the fact that, starting at the present moment, not a lot of point by point layout of the system, making it difficult to get to issues like execution, costs on (by excellence of the kind of development to be passed on) et cetera.

Different issues must be considered while doing a particular examination. Grasp the differing progressions required in the proposed system. Before starting the wander, we should be clear about what are the advances that are to be required for the change of the new system. Check whether the affiliation by and by has the required advancements. Is the required development open with the affiliation?

In case so is the utmost sufficient?

For instance – "Will the present printer have the ability to manage the new reports and structures required for the new system?"

2.3.3 OPERATIONAL FEASIBILITY

Proposed wanders are profitable just if they can be changed into information systems that will meet the affiliations working necessities. Simply communicated, this trial of probability asks with reference to whether the structure will work when it is made and presented. Are there genuine obstacles to Implementation? Here are questions that will help test the operational achievability of a wander.

- Is there sufficient help for the wander from organization from customers? In case the present structure is particularly cherished and used to the extent that individuals won't have the ability to see purposes behind change, there may be resistance.
- Are the present business methodologies qualified to the customer? If they are not, Users may welcome a change that will accomplish a more operational and supportive systems.

Have the customer been locked in with the orchestrating and change of the wander? Early commitment decreases the chances of impenetrability to the structure.

3. REQUIREMENT SPECIFICATION

The reason for this SRS record is to distinguish the necessities and functionalities for Intelligent Network Backup Tool. The SRS will characterize how our group and the customer consider the last item and the attributes or usefulness it must have. This record additionally makes a note of the discretionary prerequisites which we intend to execute yet are not required for the working of the venture.

This stage assesses the required necessities for the Images Processing for an orderly method for assessing the prerequisites a few procedures are included. The initial step associated with dissecting the prerequisites of the framework is perceiving the idea of framework for a solid examination and all the case are defined to better comprehend the investigation of the dataset.

INTENDED AUDIENCE AND READING SUGGESTIONS

This record is proposed for extend engineers, directors, clients, analyzers and documentation journalists. This report goes for examining plan and execution imperatives, conditions, framework highlights, outside interface prerequisites and other non utilitarian necessities.

IDENTIFICATION OF NEEDS

The first and imperative need for a business firm or an association is to know how they are performing in the market and parallelly they have to know how to conquer their rivals in the market.

3.1 SOFTWARE REQUIREMENTS

Operating System	:	Windows
Framework	:	Jupyter
Language	:	Python
IDE	:	Anaconda

3.2 HARDWARE REQUIREMENTS

Processor	:	Pentium 4
Hard disc	:	500GB
RAM	:	4GB

System with all standard accessories like monitor, keyboard, mouse, etc.

4. SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

4.1 SYSTEM ARCHITECTURE

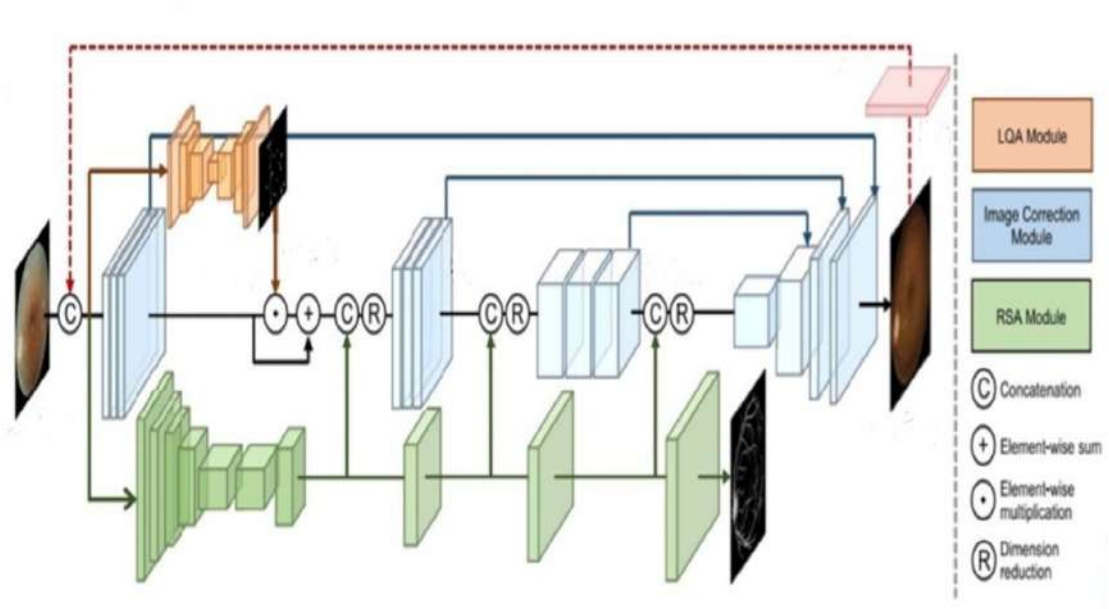


Fig 4.1 System architecture

4.2 MODULES

DATASET

The proposed method was trained with the diabetic retinopathy image dataset (DRIMDB), which is a public retinal image database. There is a total of 216 fundus images divided into three classes, good of 125, bad of 69, and outlier of 22 [11], and the outlier images were not used due to the purpose of this study only evaluate bad and good images. We collected 500 fundus images from SCH Bucheon hospital to evaluate the proposed model, and the test dataset called local datasets contains fundus images from healthy people as well as those who have glaucoma, age-related macular degeneration, and diabetic retinopathy.

DATA PREPROCESSING

Fundus images of both the training and testing dataset were resized to 299×299 since the resolution of both the training and testing datasets were different. Data augmentation was applied to the training dataset to reduce the influences of data imbalance. The methods such as zoom-in, zoom-out, vertical-flip, horizontal-flip, and contrast were applied. The total augmented images of the DRIMDB dataset (194 images) are 2,910 images which 1,875 and 1,035 are good and bad quality images, respectively.

IMAGE PRE-PROCESSING

A retinal image is composed of two major regions - a circular retina and a black background. Since the image will be filtered using a low-pass filter in the following section, unexpected results often occur when the convolution kernel slides through the retinal due to the sudden change of pixel values. Padding the black areas before an enhancement is an effective way to reduce an over enhancement of the retinal boundary. The padding can be based on a mirror reflection or content-aware filling.

CLINICAL IMAGE ANALYSIS AND APPLICATIONS

Since medical image correction models should be applied to real clinical tasks, to demonstrate the effectiveness of the proposed method, we conduct additional experiments on clinical image analysis tasks, including the vessel segmentation and optic disc/cup detection

4.3 UML DIAGRAMS

UML (Unified Modeling Language) is a standard vernacular for choosing, envisioning, making, and specifying the collectibles of programming structures. UML is a pictorial vernacular used to make programming blue prints. It is in like way used

to exhibit non programming structures similarly like process stream in a gathering unit and so forth.

UML is not a programming vernacular yet rather instruments can be utilized to make code in different tongues utilizing UML graphs. UML has an incite relationship with question composed examination and outline. UML expect a fundamental part in portraying trade viewpoints of a structure.

4.4 USE CASE DIAGRAMS

The use case graph is for demonstrating the direct of the structure. This chart contains the course of action of use cases, performing pros and their relationship. This chart might be utilized to address the static perspective of the structure.

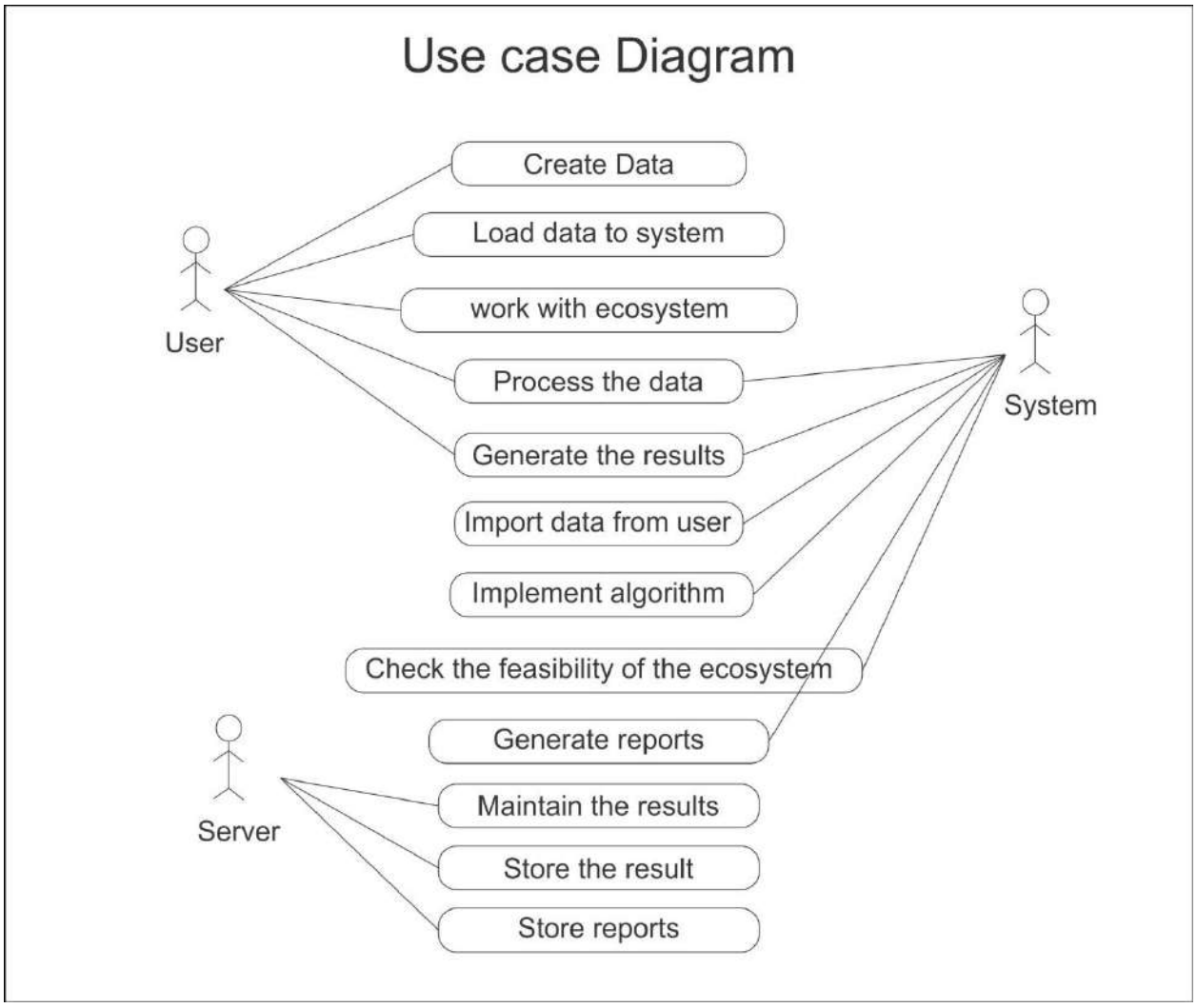


Fig:4.4 Use Case Diagram

4.5 CLASS DIAGRAM

The class graph is the most normally pulled in layout UML. It addresses the static course of action perspective of the structure. It solidifies the strategy of classes, interfaces, joint attempts and their affiliations.

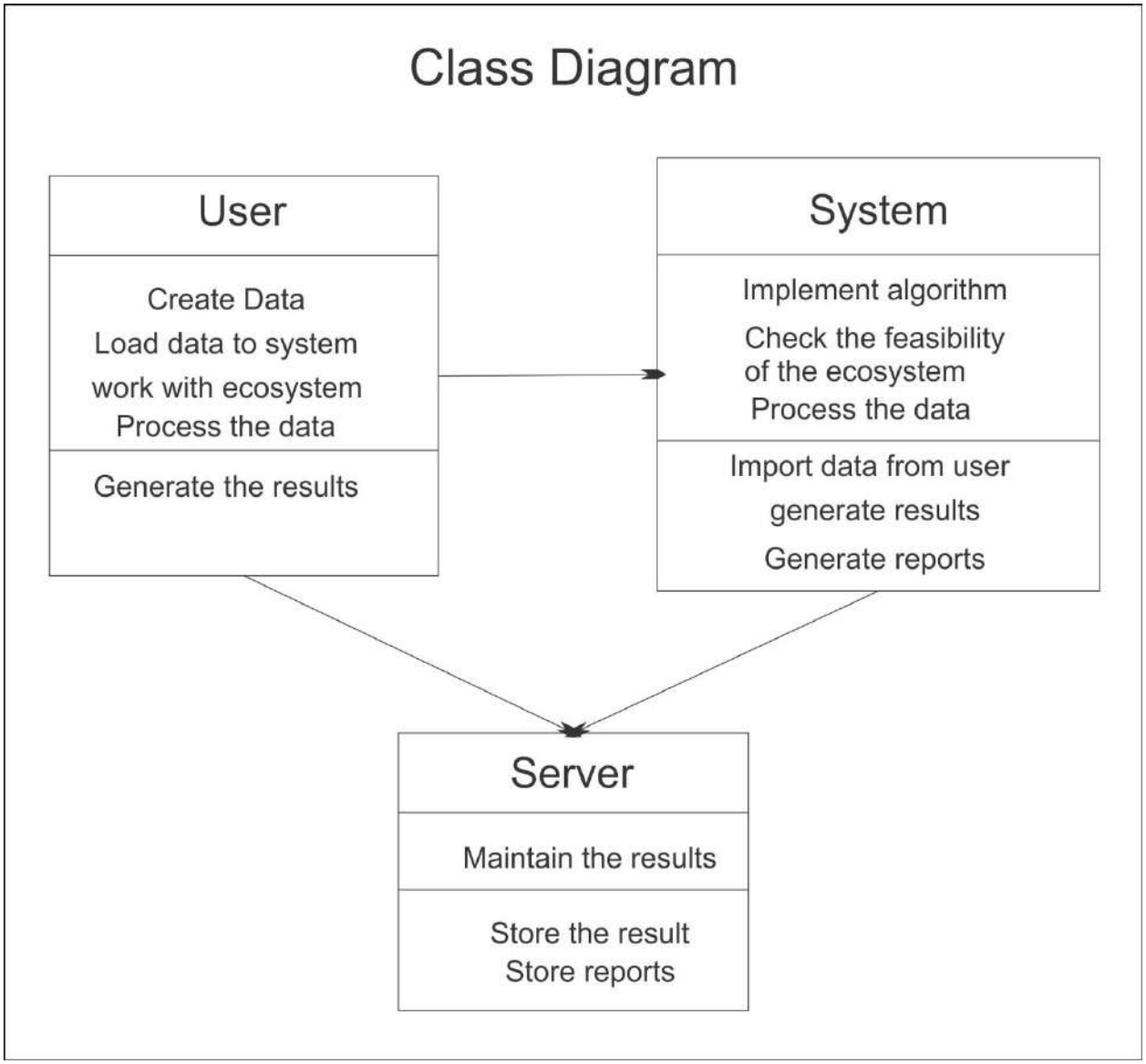


Fig:4.5 Class Diagram

In the above class diagram, the relationship that is the dependence between each one of the classes is sketched out. Additionally, even the operations performed in each and every class is similarly appeared.

4.6 SEQUENCE DIAGRAM

This is a cooperation design which tends to the time requesting of messages. It includes set of parts and the messages sent and gotten by the instance of parts. This chart is utilized to address the dynamic perspective of the structure.

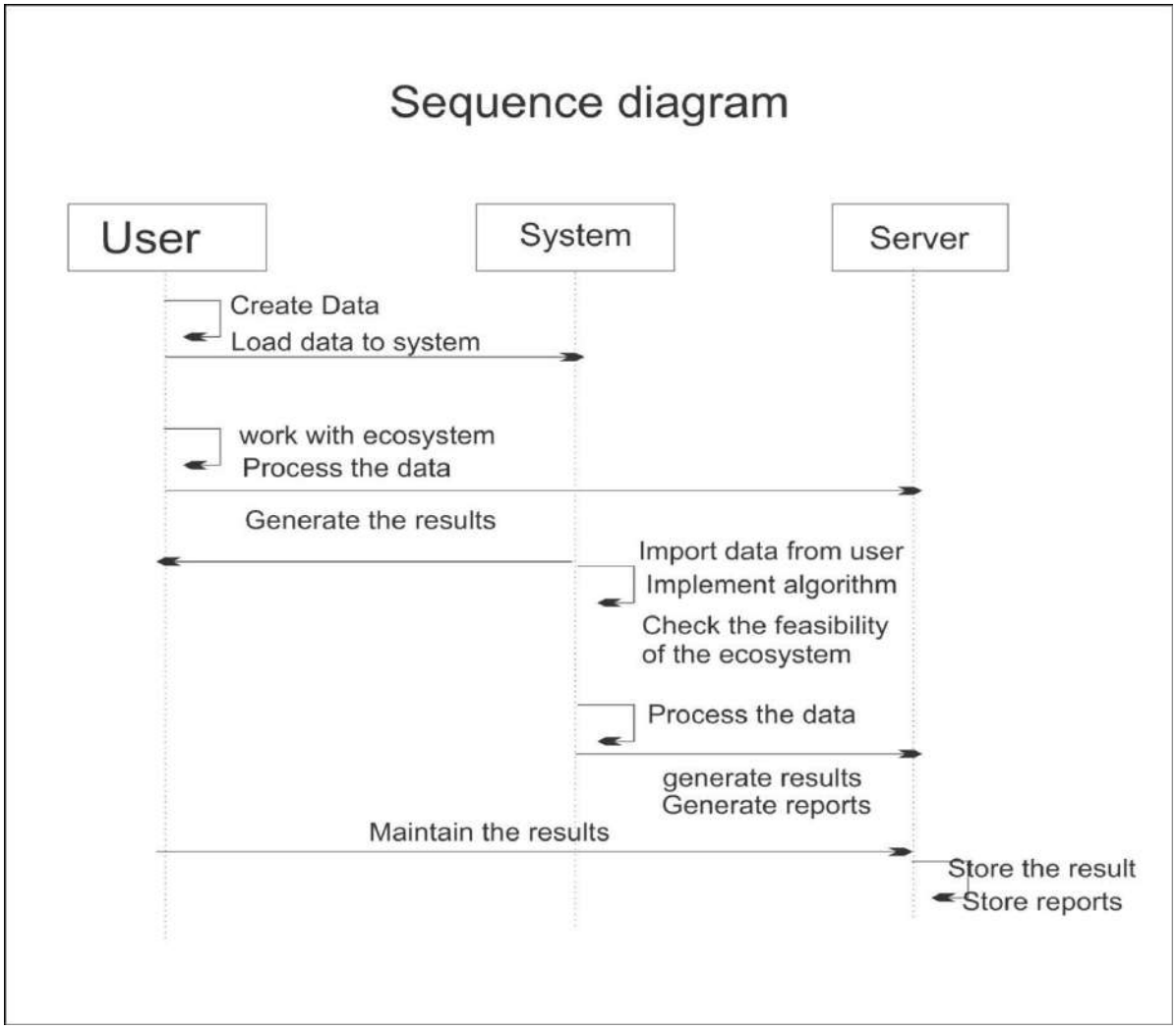


Fig:4.6 Sequence Diagram

A succession outline indicates question communications masterminded in time arrangement. In the above graph, there are five articles cooperating with each other. Each protest has a vertical dashed line which speaks to the presence of a question over some undefined time frame. This graph has additionally a tall, thin rectangle which is called center of control that demonstrates the timeframe amid which a protest is playing out an activity, either specifically or through a subordinate system.

4.7 COLLABARATION DIAGRAM

This is a support format, which tends to the principal relationship of articles that send and get messages. It incorporates set of parts, connectors that interface the parts and the messages sent and get by those parts. This graph is utilized to address the dynamic perspective of the framework.

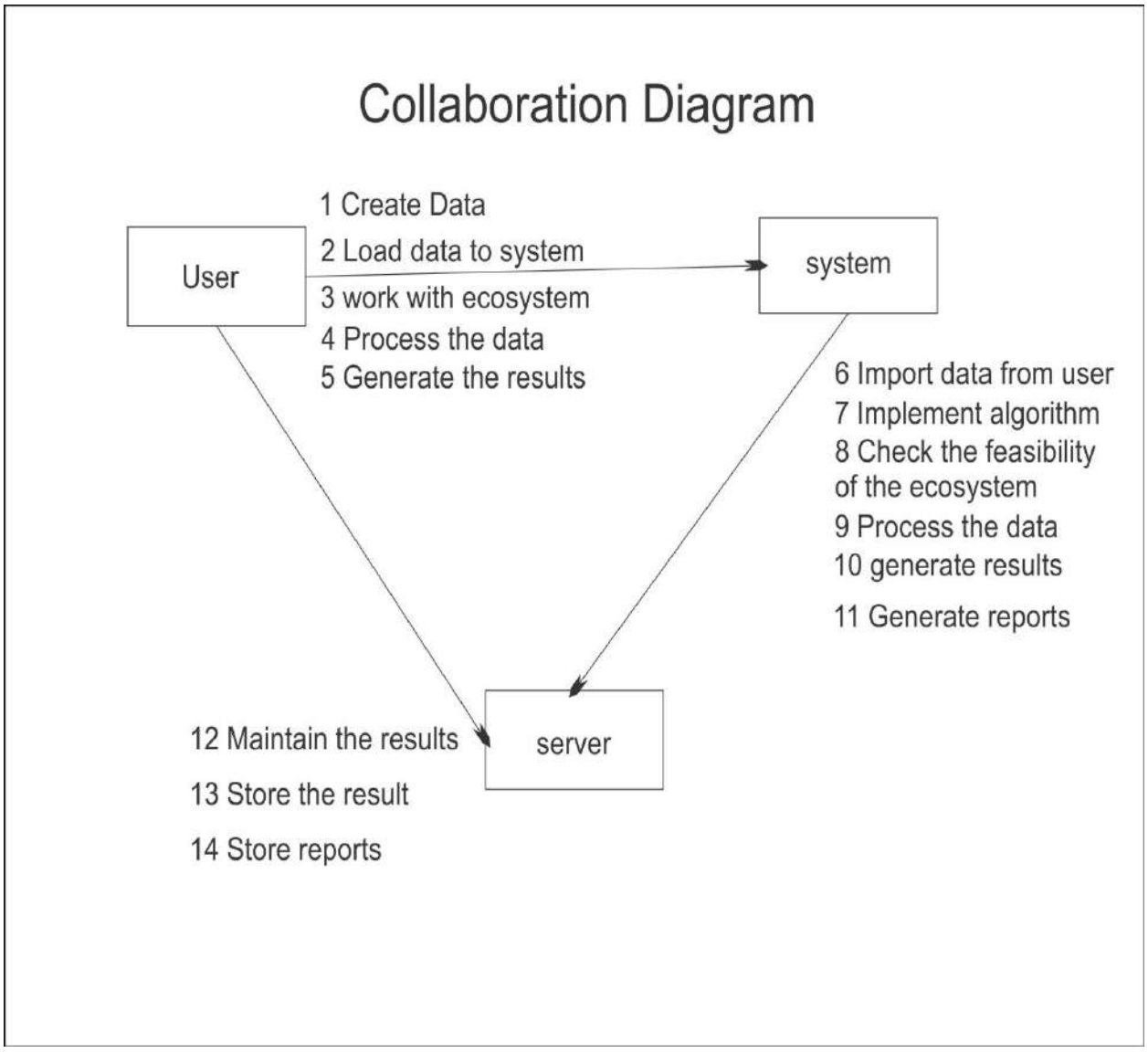


Fig:4.7 Collaboration Diagram

The joint effort outline contains articles, way and arrangement number. In the above graph, there are five questions specifically customer, client, framework, Hadoop and server. These items are connected to each other utilizing a way. A succession number show the time request of a message.

4.8 STATE CHART DIAGRAM

The state graph contains the game-plan of states, occasions and exercises. This graph is noteworthy for tending to the lead of the interface, class and made effort. The key centralization of state outline is to show the occasion sort out lead of the request. The state follows diagram the dynamic perspective of the framework.

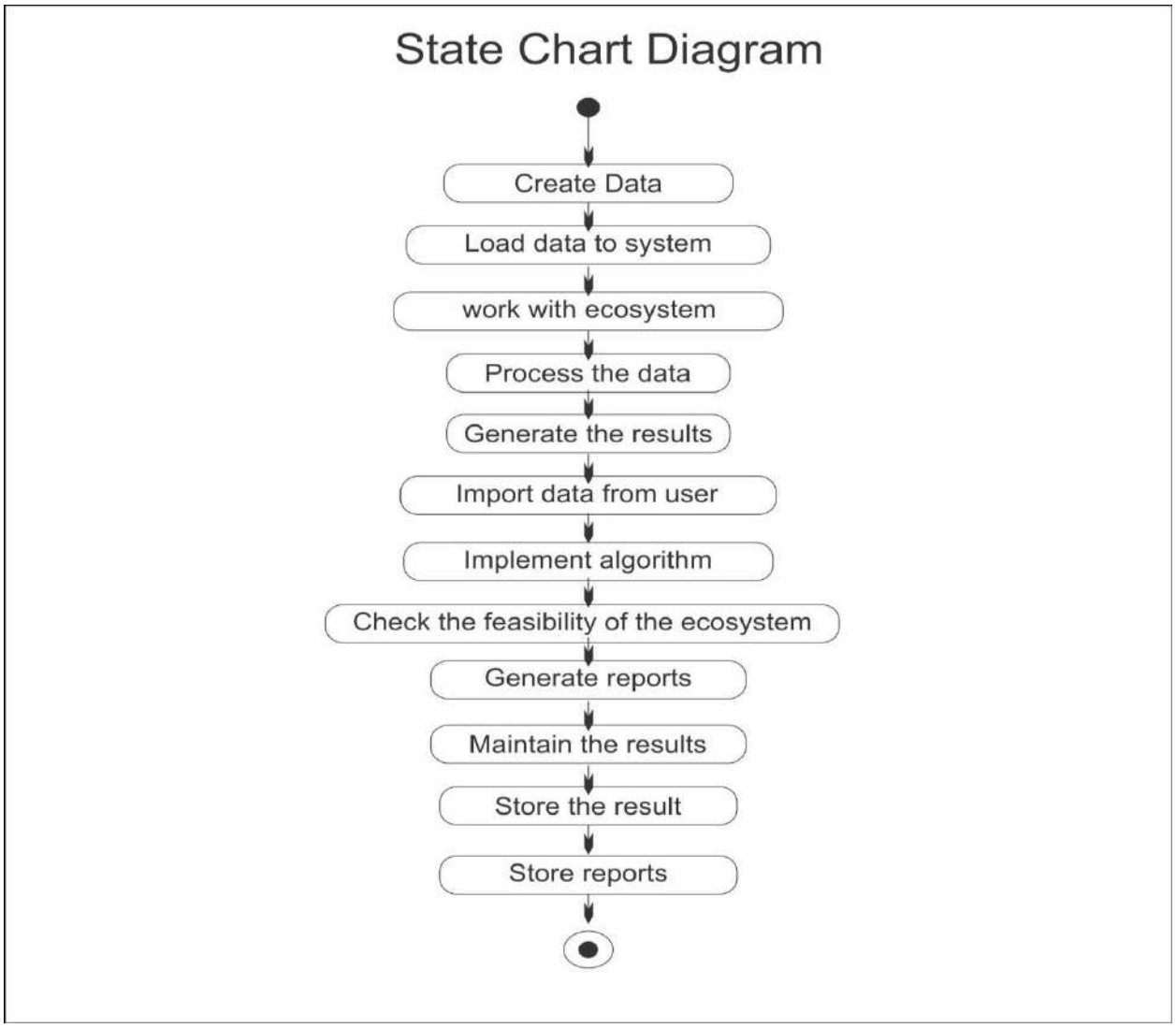


Fig:4.8 State Chart Diagram

A state outline graph contains two components called states and progress. States speak to circumstances amid the life of a question. We can without much of a stretch outline a state in Smart Draw by utilizing a rectangle with adjusted corners. Change is a strong bolt speaks to the way between various conditions of a question. Name the change with the occasion that activated it and the activity those outcomes from it.

4.9 COMPONENT DIAGRAM

The imperative portion of part format is segment. This diagram demonstrates within parts, connectors and ports that understand the piece. Precisely when section is instantiated, duplicates of inside parts are besides instantiated.

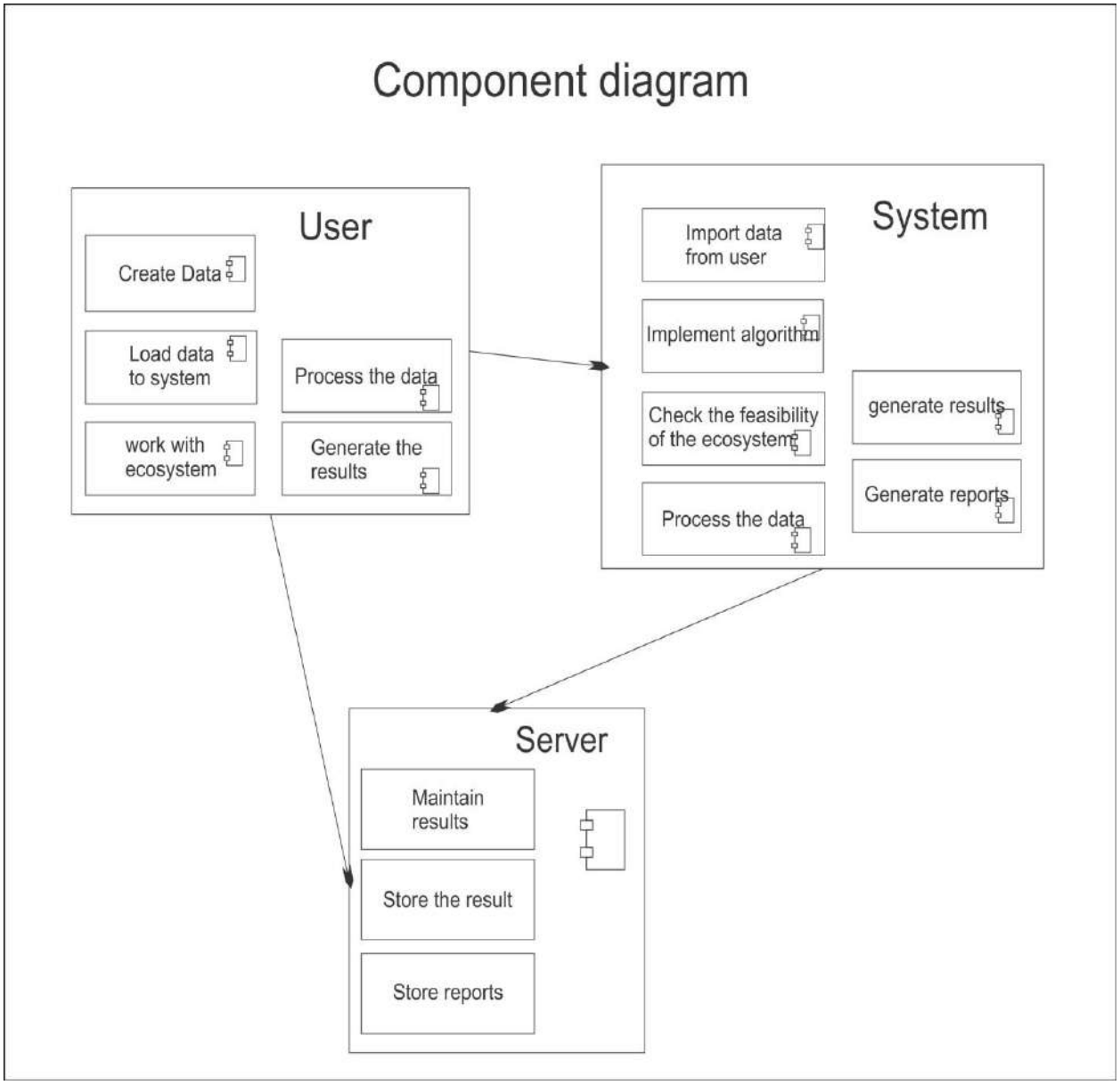


Fig:4.9 Component Diagram

A part outline is spoken to utilizing segment. A part is a physical building piece of the framework. It is spoken to as a rectangle with tab. Part outline portrays the inward handling of the venture. The information is sent to the Hadoop where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

4.10 DEPLOYMENT DIAGRAM

The fundamental fragment in game-plan layout is a middle point. The strategy of focus focuses and their relationship with other is tended to utilizing sending plot. The sending outline is identified with the area diagram, that is one focus purpose obviously of activity format frequently includes no short of what one sections. This outline is in like way critical for tending to the static perspective of the framework.

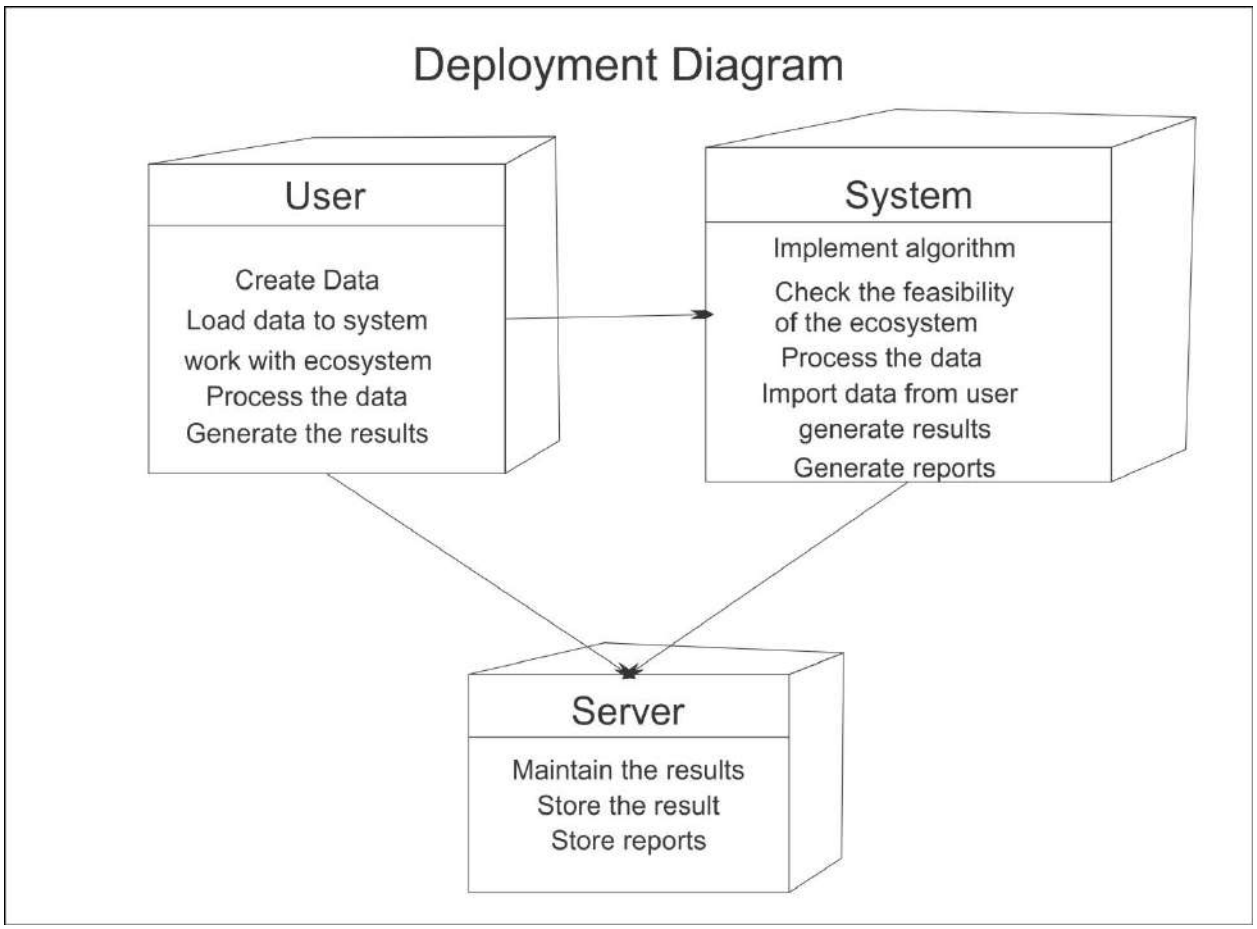


Fig:4.10 Deployment Diagram

An arrangement graph is spoken to utilizing hub. A hub is a physical asset that executes code parts. They are likewise used to portray run time handling of hubs. The information is sent to the Hadoop where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

4.11 DATA FLOW DIAGRAMS

An information stream design (DFD) is a graphical portrayal of the "stream" of information through a data framework, demonstrating its strategy edges.

Level 0: System input/ output level

A level 0 DFD describes the system wide boundaries, dealing input to and output flow from the system and major processes.

Fig 4.11.1 Level 0 DFD

Level 1: Sub system level data flow

. The Level 1 DFD exhibits how the system is secluded into sub-structures (shapes)

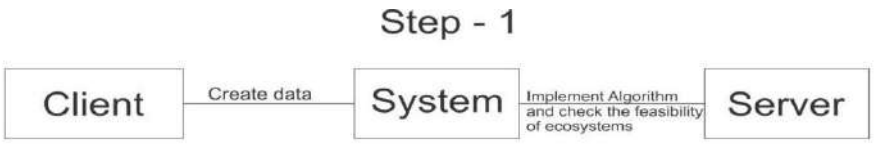


Fig 4.11.2 Level 1 DFD

Level 2: File level detail data flow

The level 2 DFD elucidates the fundamental level of understanding about the system's working.

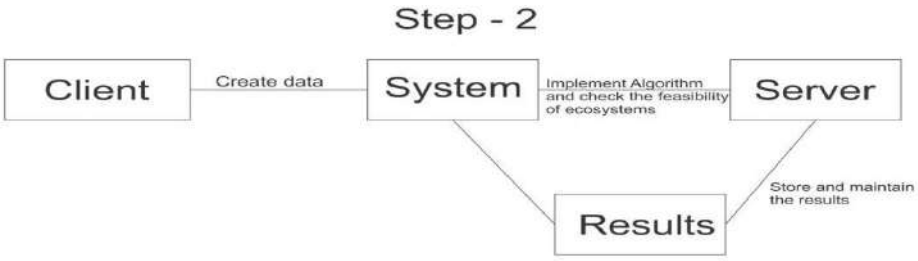


Fig 4.11.3 Level 2 DFD

5. IMPLEMENTATION

SAMPLE CODE

```
# std
import argparse
from argparse import RawTextHelpFormatter
import glob
from os import makedirs
from os.path import join, exists, basename, splitext
# 3p
import cv2
from tqdm import tqdm
# project
from exposure_enhancement import enhance_image_exposure
def main(args):
    # load images
    imdir = args.folder
    ext = ['png', 'jpg', 'bmp']    # Add image formats here
    files = []
    [files.extend(glob.glob(imdir + '*.' + e)) for e in ext]
    images = [cv2.imread(file) for file in files]
     # create save directory
    directory = join(imdir, "enhanced")
    if not exists(directory):
        makedirs(directory)
     # enhance images
    for i, image in tqdm(enumerate(images), desc="Enhancing images"):
```

```
        enhanced_image = enhance_image_exposure(image, args.gamma, args.lambda_,
not args.lime,
                                                sigma=args.sigma, bc=args.bc, bs=args.bs, be=args.be,
eps=args.eps)
        filename = basename(files[i])
        name, ext = splitext(filename)
        method = "LIME" if args.lime else "DUAL"
        corrected_name = f"{name}_{method}_g{args.gamma}_l{args.lambda_}{ext}"
        cv2.imwrite(join(directory, corrected_name), enhanced_image)
if __name__ == "__main__":
    parser = argparse.ArgumentParser(
        description="Python implementation of two low-light image enhancement
techniques via illumination map estimation.",
        formatter_class=RawTextHelpFormatter
    )
    parser.add_argument("-f", '--folder', default='./demo/', type=str,
                        help="folder path to test images.")
    parser.add_argument("-g", '--gamma', default=0.6, type=float,
                        help="the gamma correction parameter.")
    parser.add_argument("-l", '--lambda_', default=0.15, type=float,
                        help="the weight for balancing the two terms in the illumination
refinement optimization objective.")
    parser.add_argument("-ul", "--lime", action='store_true',
                        help="Use the LIME method. By default, the DUAL method is used.")
    parser.add_argument("-s", '--sigma', default=3, type=int,
                        help="Spatial standard deviation for spatial affinity based Gaussian
weights.")
    parser.add_argument("-bc", default=1, type=float,
```

```
                help="parameter for controlling the influence of Mertens's contrast
measure.")
    parser.add_argument("-bs", default=1, type=float,
                help="parameter for controlling the influence of Mertens's saturation
measure.")
    parser.add_argument("-be", default=1, type=float,
                help="parameter for controlling the influence of Mertens's well
exposedness measure.")
    parser.add_argument("-eps", default=1e-3, type=float,
                help="constant to avoid computation instability.")


    args = parser.parse_args()
    main(args)
```

6. TESTING

6.1 INTRODUCTION TO TESTING

Testing is a procedure, which uncovers blunders in the program. Programming testing is a basic component of programming quality affirmation and speaks to a definitive audit of determination, outline and coding. The expanding perceivability of programming as a framework component and chaperon costs related with a product disappointment are propelling variables for we arranged, through testing. Testing is the way toward executing a program with the plan of finding a mistake. The plan of tests for programming and other built items can be as trying as the underlying outline of the item itself It is the significant quality measure utilized amid programming improvement. Amid testing, the program is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing as it is relied upon to perform.

6.2 TESTING STRATEGIES

A technique for programming testing coordinates the outline of programming experiments into an all around arranged arrangement of steps that outcome in fruitful improvement of the product. The procedure gives a guide that portrays the means to be taken, when, and how much exertion, time, and assets will be required. The procedure joins test arranging, experiment configuration, test execution, and test outcome gathering and assessment. The procedure gives direction to the specialist and an arrangement of points of reference for the chief. Due to time weights, advance must be quantifiable and issues must surface as ahead of schedule as would be prudent

Keeping in mind the end goal to ensure that the framework does not have blunders, the distinctive levels of testing techniques that are connected at varying periods of programming improvement are:

6.3 UNIT TESTING

Unit Testing is done on singular modules as they are finished and turned out to be executable. It is restricted just to the planner's prerequisites. It centers testing around the capacity or programming module. It Concentrates on the interior preparing

rationale and information structures. It is rearranged when a module is composed with high union

- Reduces the quantity of experiments
- Allows mistakes to be all the more effectively anticipated and revealed

6.4 BLACK BOX TESTING

It is otherwise called Functional testing. A product testing strategy whereby the inward workings of the thing being tried are not known by the analyzer. For instance, in a discovery test on a product outline the analyzer just knows the information sources and what the normal results ought to be and not how the program touches base at those yields. The analyzer does not ever inspect the programming code and does not require any further learning of the program other than its determinations. In this system some experiments are produced as information conditions that completely execute every single practical prerequisite for the program. This testing has been utilizations to discover mistakes in the accompanying classifications:

- Incorrect or missing capacities
- Interface blunders
- Errors in information structure or outside database get to
- Performance blunders
- Initialization and end blunders.

In this testing just the yield is checked for rightness.

6.5 WHITE BOX TESTING

It is otherwise called Glass box, Structural, Clear box and Open box testing . A product testing procedure whereby express learning of the inner workings of the thing being tried are utilized to choose the test information. Not at all like discovery testing, white box testing utilizes particular learning of programming code to inspect yields. The test is precise just if the analyzer comprehends what the program should do. He or she would then be able to check whether the program veers from its

expected objective. White box testing does not represent blunders caused by oversight, and all obvious code should likewise be discernable. For an entire programming examination, both white box and discovery tests are required.

In this the experiments are produced on the rationale of every module by drawing stream diagrams of that module and sensible choices are tried on every one of the cases. It has been utilizations to produce the experiments in the accompanying cases:

- Guarantee that every single free way have been Executed.
- Execute every single intelligent choice on their actual and false Sides.

6.6 INTEGRATION TESTING

Coordination testing guarantees that product and subsystems cooperate an entirety. It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime reconciliation is happening, lead tests to reveal blunders related with interfaces. Its Objective is to take unit tried modules and assemble a program structure in view of the recommended outline

Two Approaches of Integration Testing

- Non-incremental Integration Testing
- Incremental Integration Testing

6.7 SYSTEM TESTING

Framework testing includes in-house testing of the whole framework before conveyance to the client. Its point is to fulfill the client the framework meets all necessities of the customer's determinations. This testing assesses working of framework from client perspective, with the assistance of particular report. It doesn't require any inward learning of framework like plan or structure of code.

It contains utilitarian and non-useful zones of utilization/item. Framework Testing is known as a super arrangement of a wide range of testing as all the significant sorts of testing are shrouded in it. In spite of the fact that attention on sorts of testing may differ on the premise of item, association procedures, course of events and necessities. Framework Testing is the start of genuine testing where you test an item all in all and not a module/highlight.

6.8 ACCEPTANCE TESTING

Acknowledgment testing, a testing method performed to decide if the product framework has met the prerequisite particulars. The principle motivation behind this test is to assess the framework's consistence with the business necessities and check in the event that it is has met the required criteria for conveyance to end clients. It is a pre-conveyance testing in which whole framework is tried at customer's site on genuine information to discover blunders. The acknowledgment test bodies of evidence are executed against the test information or utilizing an acknowledgment test content and afterward the outcomes are contrasted and the normal ones.

The acknowledgment test exercises are completed in stages. Right off the bat, the essential tests are executed, and if the test outcomes are palatable then the execution of more intricate situations are done.

6.9 TEST APPROACH

A Test approach is the test system usage of a venture, characterizes how testing would be done. The decision of test methodologies or test technique is a standout amongst the most intense factor in the achievement of the test exertion and the precision of the test designs and gauges.

Testing should be possible in two ways

- Bottom up approach
- Top down approach

BOTTOM UP APPROACH

Testing can be performed beginning from littlest and most reduced level modules and continuing each one in turn. In this approach testing is directed from sub module to primary module, if the fundamental module is not built up a transitory program called DRIVERS is utilized to recreate the principle module. At the point when base level modules are tried consideration swings to those on the following level that utilization the lower level ones they are tried exclusively and afterward connected with the already inspected bring down level modules

TOP DOWN APPROACH

In this approach testing is directed from fundamental module to sub module. in the event that the sub module is not built up an impermanent program called STUB is utilized for mimic the sub module. This sort of testing begins from upper level modules. Since the nitty gritty exercises more often than not performed in the lower level schedules are not given stubs are composed. A stub is a module shell called by upper level module and that when achieved legitimately will restore a message to the calling module demonstrating that appropriate association happened.

6.10 VALIDATION

The way toward assessing programming amid the improvement procedure or toward the finish of the advancement procedure to decide if it fulfills determined business prerequisites. Approval Testing guarantees that the item really addresses the customer's issues. It can likewise be characterized as to exhibit that the item satisfies its proposed utilize when sent on proper condition.

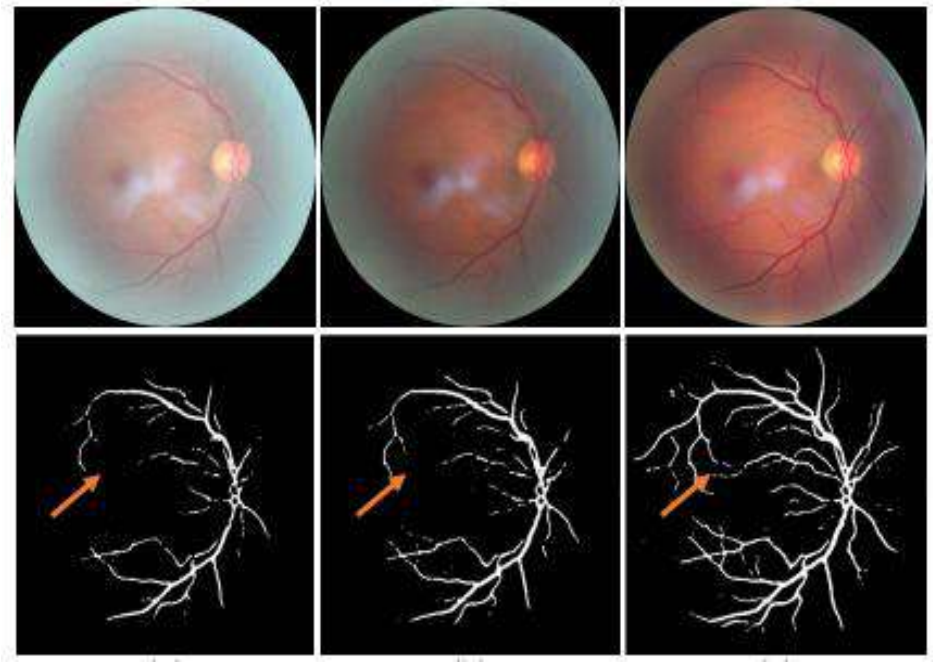
The framework has been tried and actualized effectively and along these lines guaranteed that every one of the prerequisites as recorded in the product necessities determination are totally satisfied.

6.11 TEST CASES

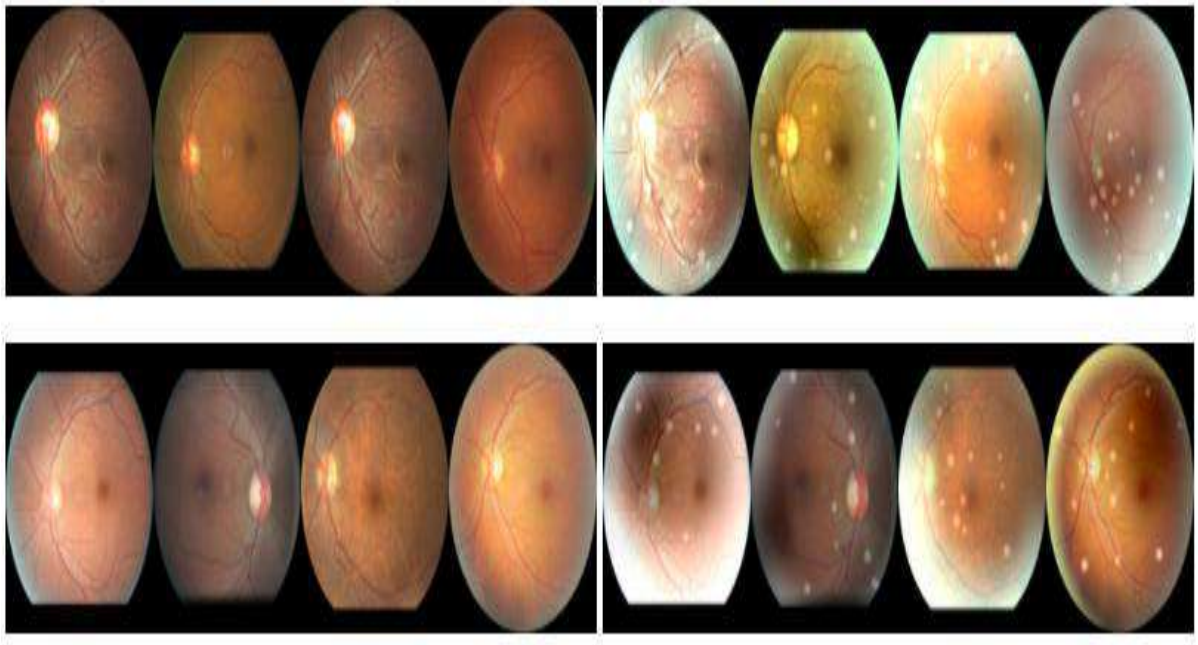
Experiments include an arrangement of steps, conditions and sources of info that can be utilized while performing testing undertakings. The principle expectation of this action is to guarantee whether a product passes or bombs as far as usefulness

and different perspectives. The way toward creating experiments can likewise help discover issues in the prerequisites or plan of an application. Experiment goes about as the beginning stage for the test execution, and in the wake of applying an arrangement of information esteems, the application has a conclusive result and leaves the framework at some end point or otherwise called execution post condition.

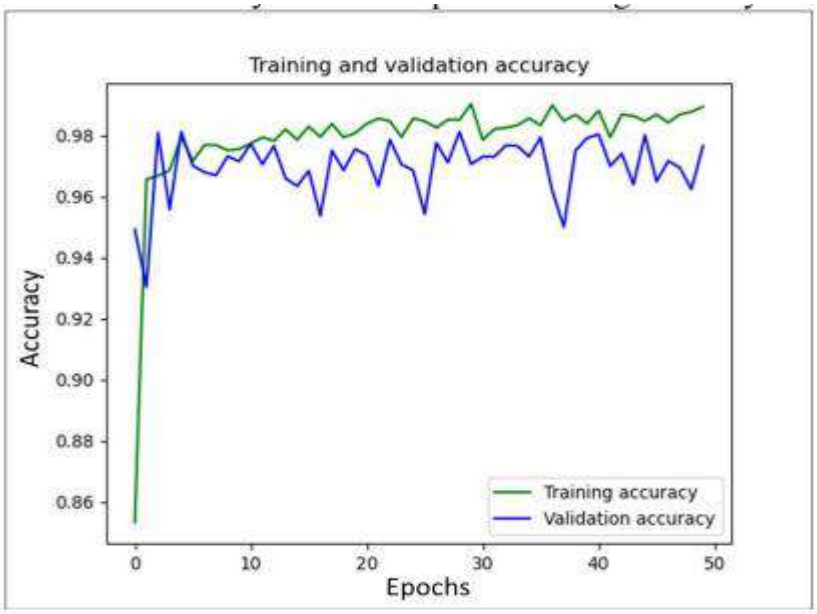
7. SCREENSHOTS



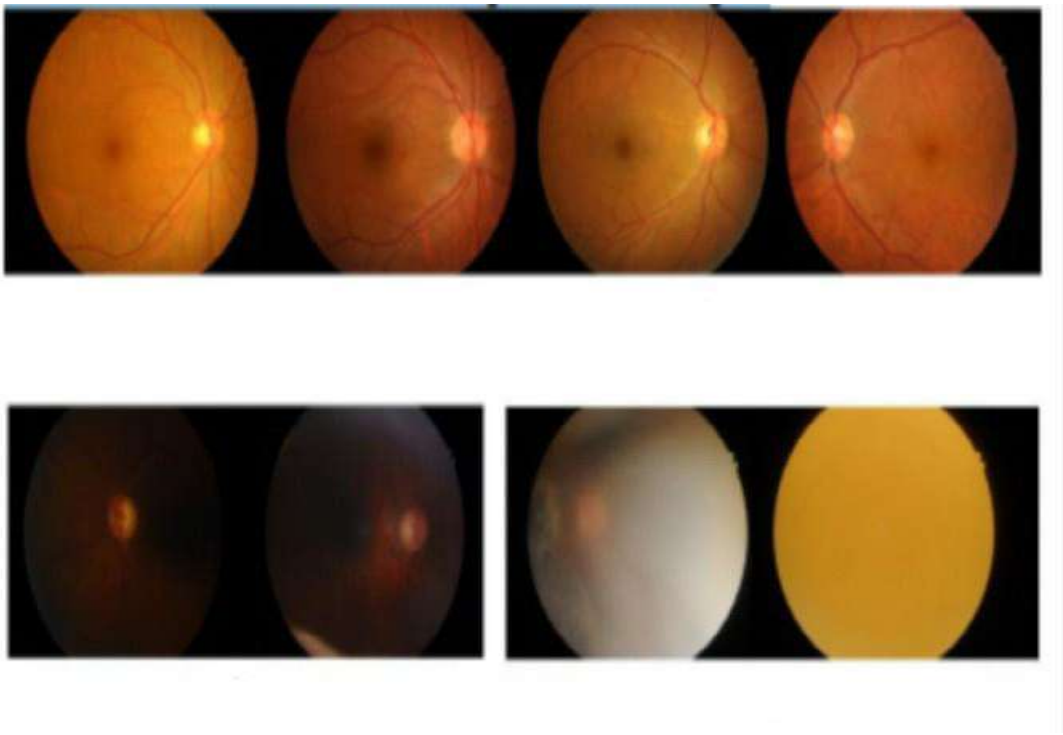
Screenshot 7.1 : Illustration of fundus image correction



Screenshot 7.2 : Corrected images



Screenshot 7.3 :Accuracy and Error curves



Screenshot 7.4 :Example of good images predicted by the proposed method

8. CONCLUSION

The local dataset contained 500 retinal images have evaluated by our system and performed the results that contained 482 images as good, and 18 images as bad quality images. Fig. shows the example of good images from a prediction by our system. According to Migual et al. [12], the effect from uneven illumination (over dark and overbright) over the macula, uneven illumination of the optic disc, uneven illumination of the image edge cause performance of diagnosis disease.

9.FUTURE SCOPE

We have proposed a clinically oriented fundus enhancement network, named cofe-Net, to correct low-quality fundus image while preserving accurate lesion areas and retinal structures. Furthermore, a complete degradation model has also been introduced to generate adequate training image pairs. Experiments support our insight into the problems of fundus image correction and degradation factor modeling. Our cofeNet can boost the performance for different clinical tasks, such as vessel segmentation and disc/cup detection. Our method can also assist ophthalmologists in ocular disease diagnosis through retinal fundus image observation and analysis, while also being beneficial to automated image analysis systems.

10. REFERENCE

[1] M. D. Abramoff, M. K. Garvin, and M. Sonka, “Retinal Imaging and Image Analysis,” IEEE Reviews in Biomedical Engineering, vol. 3, pp. 169–208, 2010.

[2] U. Schmidt-Erfurth, A. Sadeghipour, B. S. Gerendas, S. M. Waldstein, and H. Bogunovic, “Artificial intelligence in retina,” ´ Progress in Retinal and Eye Research, vol. 67, pp. 1–29, 2018.

[3] S. Philip, L. M. Cowie, and J. A. Olson, “The impact of the Health Technology Board for Scotland’s grading model on referrals to ophthalmology services,” British Journal of Ophthalmology, vol. 89, no. 7, pp. 891–896, 2005.

[4] E. Peli and T. Peli, “Restoration of retinal images obtained through cataracts,” IEEE Transactions on Medical Imaging, vol. 8, no. 4, pp. 401–406, 1989.

[5] H. Fu, B. Wang, J. Shen, S. Cui, Y. Xu, J. Liu, and L. Shao, “Evaluation of Retinal Image Quality Assessment Networks in Different ColorSpaces,” in MICCAI, 2019, pp. 48–56.

[6] Y. Cheng, F. Juefei-Xu, Q. Guo, H. Fu, X. Xie, S.-W. Lin, W. Lin, and Y. Liu, “Adversarial Exposure Attack on Diabetic Retinopathy Imagery,” arXiv, 2020.

[7] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, S. Gao, and J. Liu, “CE-Net: Context Encoder Network for 2D Medical Image Segmentation,” IEEE Transactions on Medical Imaging, vol. 38, no. 10, pp. 2281–2292, 2019.

[8] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, “HDR image reconstruction from a single exposure using deep CNNs,” ACM TOG (Proceedings of SIGGRAPH), vol. 36, no. 6, p. 178, 2017.

[9] X. Yang, K. Xu, Y. Song, Q. Zhang, X. Wei, and R. W. H. Lau, “Image Correction via Deep Reciprocating HDR Transformation,” in CVPR, 2018, pp. 1798–1807.

[10] W. Ren, S. Liu, L. Ma, Q. Xu, X. Xu, X. Cao, J. Du, and M. Yang, “Low-Light Image Enhancement via a Deep Hybrid Network,” IEEE Transactions on Image Processing, vol. 28, no. 9, pp. 4364–4375, 2019.