Eye Disease Detection using Machine Learning

Abstract:The dominant causes of visual impairment worldwide are Cataract, Glaucoma, and retinal diseases among patients. The alarming cases of these diseases call for an urgent intervention by early diagnosis. The proposed system is designed and developed to easily facilitate the detection of cataract, glaucoma and retinal diseases among patients. The Logistic Regression, Random Forest, Gradient Boosting and Support Vector Machine algorithms are used for detection. Now days a lot of people having eye disease problem and for knowing their disease they do have to wait a lot because of the machine system in the hospital. For resolving that issue, we have developed eye disease detection model using machine learning technology which will help the patient to know their disease as early as they can. The eye disease detection model is trained on huge number parameter so that can predict the eye disease quickly.

Keywords: Eye disease, Keras, Machine learning, Convolution neural network, Supervised Learning

1.INTRODUCTION

The Eye disease detection machine learning project is developed for disease detection using machine learning. In our project we categorized the disease in 3 phase that is normal images which has no disease and second is glaucoma and third is retinopathy. So we have taken left and right side images of each class and trained our data with the help of 372 images so that our model can learn from it and converted the input dataset in grayscale format for better understanding of its feature. The dataset is converted with help of one software that is Irfan view. For the eye disease detection, we just have created model with help of keras and some of the machine learning concept. The dominant causes of visual impairment worldwide are Glaucoma, and retinal diseases among patients. The proposed system is designed and developed to easily facilitate the detection of glaucoma and retinal diseases among patients. The CNN used for detection [1].

2.LITERATURE SURVEY

2.1 Wavelet-Based Energy Features for Glaucomatous Image Classification

AUTHORS: Sumeet Dua; U. Rajendra Acharya; Pradeep Chowriappa; S. Vinitha Sree

ABSTRACT: Texture features within images are actively pursued for accurate and efficient glaucoma classification. Energy distribution over wavelet subbands is applied to find these important texture features. In this paper, we investigate the discriminatory potential of wavelet features obtained from the daubechies (db3), symlets (sym3), and biorthogonal (bio3.3, bio3.5, and bio3.7) wavelet filters. We propose a novel technique to extract energy signatures obtained using 2-D discrete wavelet transform, and subject these signatures to different feature ranking and feature selection strategies. We have gauged the effectiveness of the resultant ranked and selected subsets of features using a support vector machine, sequential minimal optimization, random forest, and naïve Bayes classification strategies. We observed an accuracy of around 93% using tenfold cross validations to demonstrate the effectiveness of these methods.

2.2 Automated diagnosis of glaucoma using digital fundus images

AUTHORS: Jagadish Nayak, Rajendra Acharya U, P Subbanna Bhat, Nakul Shetty, Teik-Cheng Lim

ABSTRACT: Glaucoma is a disease of the optic nerve caused by the increase in the intraocular pressure of the eye. Glaucoma mainly affects the optic disc by increasing the cup size. It can lead to the blindness if it is not detected and treated in proper time. The detection of glaucoma through Optical Coherence Tomography (OCT) and Heidelberg Retinal Tomography (HRT) is very expensive. This paper presents a novel method for glaucoma detection using digital fundus images. Digital image processing techniques, such as preprocessing, morphological operations and thresholding, are widely used for the

automatic detection of optic disc, blood vessels and computation of the features. We have extracted features such as cup to disc (c/d) ratio, ratio of the distance between optic disc center and optic nerve head to diameter of the optic disc, and the ratio of blood vessels area in inferior-superior side to area of blood vessel in the nasal-temporal side. These features are validated by classifying the normal and glaucoma images using neural network classifier. The results presented in this paper indicate that the features are clinically significant in the detection of glaucoma. Our system is able to classify the glaucoma automatically with a sensitivity and specificity of 100% and 80% respectively.

2.3 Image processing based automatic diagnosis of glaucoma using wavelet features of segmented optic disc from fundus image

AUTHORS: A. Singh, MK. Dutta, MP. Sarathi, V. Uher and R. Burget

ABSTRACT: Glaucoma is a disease of the retina which is one of the most common causes of permanent blindness worldwide. This paper presents an automatic image processing based method for glaucoma diagnosis from the digital fundus image. In this paper wavelet feature extraction has been followed by optimized genetic feature selection combined with several learning algorithms and various parameter settings. Unlike the existing research works where the features are considered from the complete fundus or a sub image of the fundus, this work is based on feature extraction from the segmented and blood vessel removed optic disc to improve the accuracy of identification. The experimental results presented in this paper indicate that the wavelet features of the segmented optic disc image are clinically more significant in comparison to features of the whole or sub fundus image in the detection of glaucoma from fundus image. Accuracy of glaucoma identification achieved in this work is 94.7% and a comparison with existing methods of glaucoma detection from fundus image indicates that the proposed approach has improved accuracy of classification.

2.4 Applications of deep learning in fundus images: A review

AUTHORS: T. Li, W. Bo, C. Hu, H. Kang, H. Liu, K. Wang, H. Fu

ABSTRACT: The use of fundus images for the early screening of eye diseases is of great clinical importance. Due to its powerful performance, deep learning is becoming more and more popular in related applications, such as lesion segmentation, biomarkers segmentation, disease diagnosis and image synthesis. Therefore, it is very necessary to summarize the recent developments in deep learning for fundus images with a review paper. In this review, we introduce 143 application papers with a carefully designed hierarchy. Moreover, 33 publicly available datasets are presented. Summaries and analyses are provided for each task. Finally, limitations common to all tasks are revealed and possible solutions are given. We will also release and regularly update the state-of-the-art results and newly-released

2.5 Faster R-CNN classification for the recognition of glaucoma

AUTHORS: S. Ajitha and MV. Judy

ABSTRACT: Glaucoma is an optic neuropathy characterized by progressive degeneration of retinal ganglion cells. The early identification of Glaucoma is extremely important as it is detrimental to one's blindness. In this paper, we present the identification of glaucoma using faster R-CNN which is one of the most well-known object detection neural networks. The proposed method uses artificial intelligence and enhanced deep learning to detect Glaucoma. Faster R-CNN comprises two modules, the region proposal network (RPN), in which the region of object is distinguished on the picture, and a network that enables to classify the objects in the proposed region. We have accomplished the finest output by applying a transfer learning scheme with ResNet50 and VGG16. Using ResNet we have detected Glaucoma with up to 96% accuracy. The test results obtained by making use of two unique publicly available data sets DRISHTI_GS and ORIGA with 751 images demonstrate that this arrangement can be a significant alternative for the computer design aid framework for the large-scale screening programs of glaucoma detection

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

In existing system, accuracy of algorithms and SVM classifiers based upon the glaucoma, retina, cataract and normal eye's fundus images is less. The Logistic Regression, Random Forest, Gradient Boosting and Support Vector Machine algorithms are used for detection.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- It SVM accuracy is very less.

3.2PROPOSED SYSTEM:

we have developed eye disease detection model using machine learning technology which will help the patient to know their disease as early as they can. The eye disease detection model is trained on huge number parameter so that can predict the eye disease quickly.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM:

- It also shows better performance compared to another method that utilized deep learning models for feature extraction and Support Vector Machine (SVM) for classification.

3.3 SYSTEM REQUIREMENTS:

SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idel 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupiter (or)**
- **Google colab**

HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system : windows, linux**
- **Processor : minimum intel i3**
- **Ram : minimum 4 gb**
- **Hard disk : minimum 250gb**

3.4 FUNCTIONAL REQUIREMENTS

1.Data Collection

2.Data Preprocessing

3.Training And Testing

4.Modiling

5.Predicting

3.5 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

3.6 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the

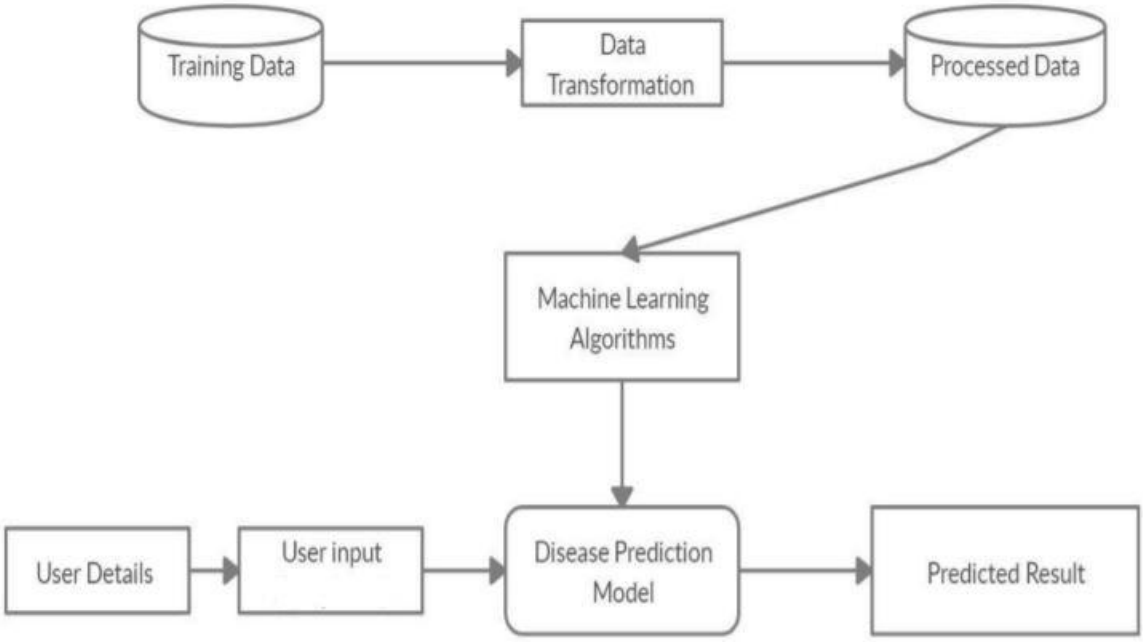
available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:



4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major

components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

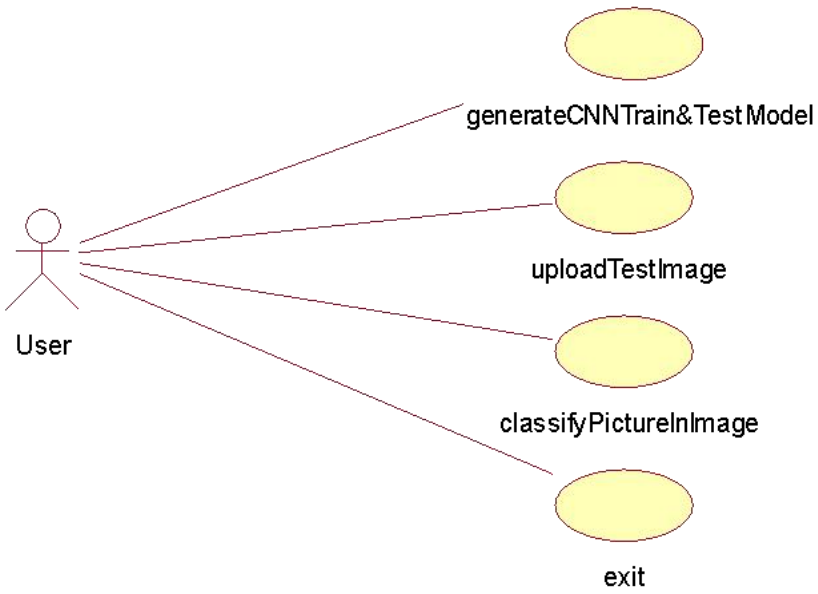
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.3.1Use case diagram:

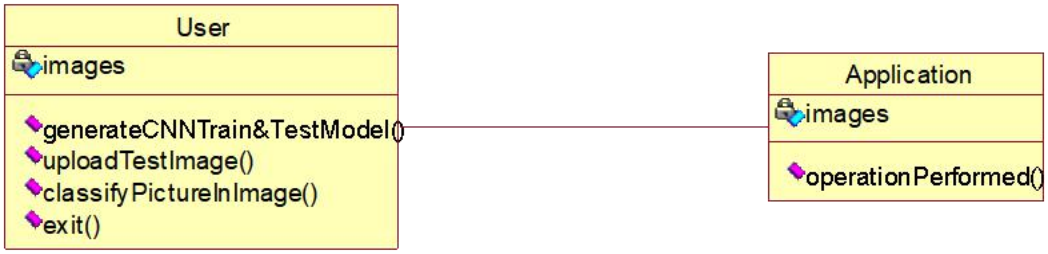
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals

(represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



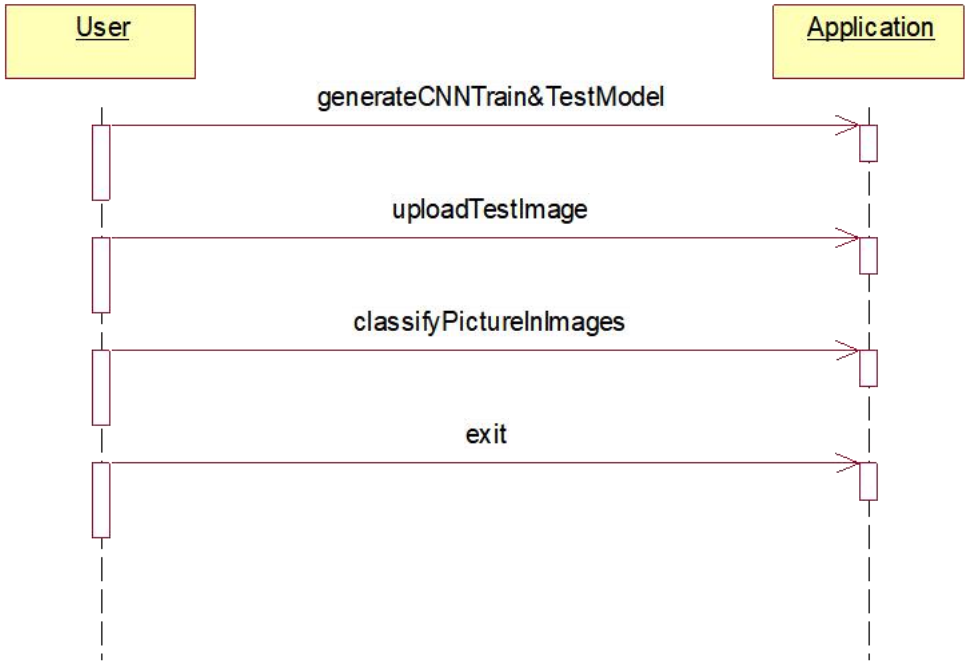
4.3.2 Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



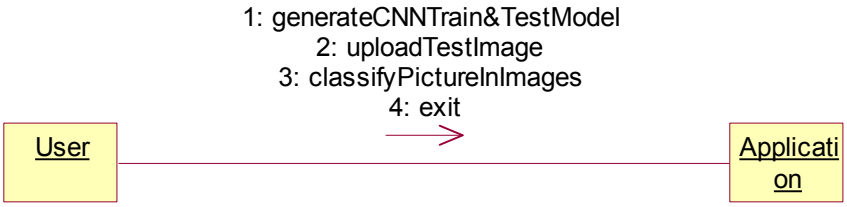
4.3.6 Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



4.3.7 Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.



4.4 IMPLEMENTATION:

MODULES:

1.Data Collection

This step was done by the original fundus of the dataset. And the composition of the dataset.understand the relationship among different features. A plot of the core features and the entire dataset.The dataset is further split into 2/3 for training and 1/3 for testing the algorithms. Furthermore, in order to obtain a representative sample, each class in the full dataset is represented in about the right proportion in both the training and testing datasets. The various proportions of the training and testing datasets used in the paper.

2.Data Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency o the algorithm. The outliers have to be removed and also variable conversion need to be done. In order to overcoming these issues we use map function.

3.Model Selection

Deep learning is about predicting and recognizing patterns and generate suitable results after understanding them. DL algorithms study patterns in data and learn from them. An DL model will learn and improve on each attempt. To gauge the effectiveness of a model, it's vital to split the data into training and test sets first. So before training our models, we split the data into Training set which was 70% of the whole dataset and Test set which was the remaining 30%. Then it was important to implement a selection of performance metrics to the predictions made by our model. In this case, we tried to identify whether an individual is going to default on a loan or not. Model accuracy might not be the sole metric to identify how our model performed- the F1 score and confusion matrix should be important metrics to analyse as well. What is important is that the right performance measures are chosen for the right situations.

4.Predict the results

The designed system is tested with test set and the performance is assured. Evolution analysis refers to the description and model regularities or trends for objects whose behavior changes over time. Common metrics calculated from the confusion matrix are Precision; Accuracy. The mot important features since these features are to develop a predictive model using ordinary CNN Classifier model.

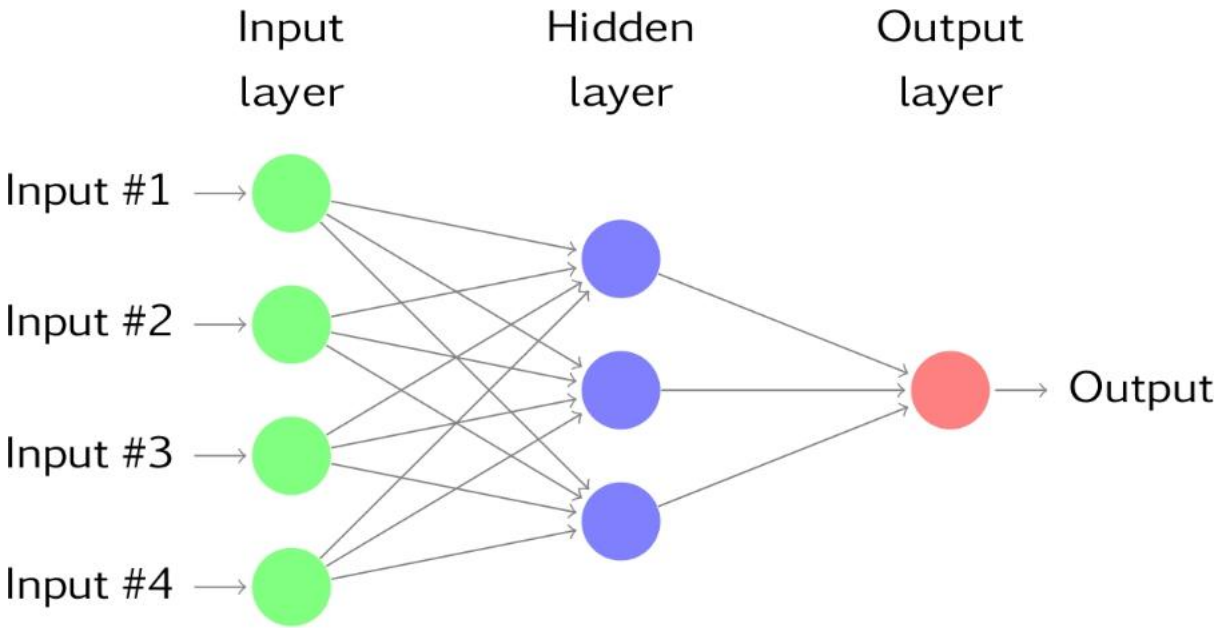
4.5 ALGORITHMS:

CNN ALGORITHM:

To demonstrate how to build a convolutional neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. This network that we shall build is a very small network that we can run on a CPU as well. Traditional neural networks that are very good at doing image classification have many more parameters and take a lot of time if trained on normal CPU. However, our objective is to show how to build a real-world convolutional neural network using TENSORFLOW.

Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on it (say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx + b). This value is passed to a non-linear function called activation function (f) to produce the final output (activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid. The neuron which uses sigmoid function as an activation function will be called sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH.

If you stack neurons in a single line, it's called a layer; which is the next building block of neural networks. See below image with layers



To predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left.

5.SOFTWARE ENVIRONMENT

What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language

that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Deep Neural Networks

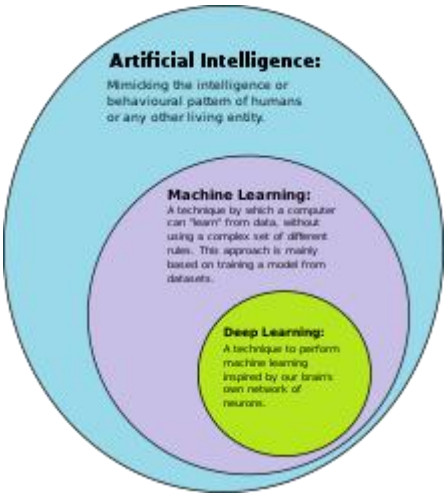
Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, artificial neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.

The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a nonpolynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which

is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part.



Deep learning creates many layers of neurons, attempting to learn structured representation of big data, layer by layer.

Architecture of the network:

Network models

• Deep neural networks are mathematical models of intelligence designed to mimic human brains.

• Network models define a set of network layers and how they interact.

• Questions to answer while designing a network models include: –

Which layer type to use? – How many neurons to use in each layer? – How are layers arranged? – And more

• There are many standard CNN models available today which work great for many standard problems. Examples being AlexNet, GoogleNet, Inception-ResNet, VGG, etc

Deep learning frameworks

• Building a deep learning solution is a big challenge because of its complexity.

• Frameworks are tools to ease the building of deep learning solutions.

• Frameworks offer a higher level of abstraction and simplify potentially difficult programming tasks.

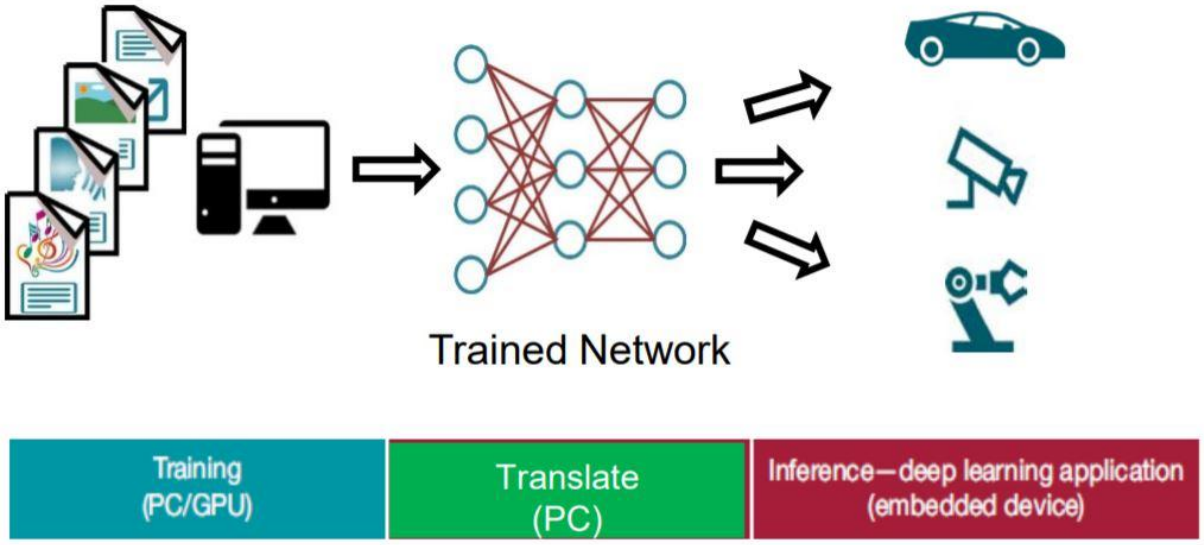
Popular Frameworks:

• TensorFlow: – Developed by Google – The most used deep learning framework – Based on Github stars and forks and Stack Overflow activity

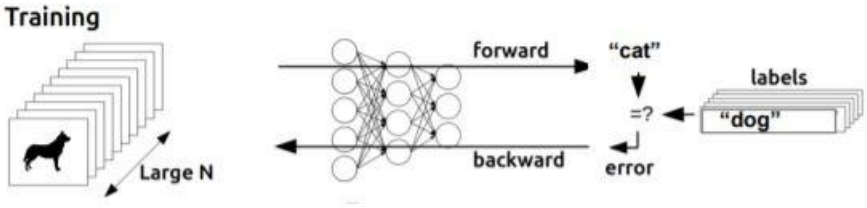
• Caffe: – Developed by Berkeley Vision and Learning Center (BVLC) – Popular for CNN modeling (imaging/computer vision applications) and its Model Zoo (a selection of pre-trained networks) Next to all these frameworks, there are also interfaces that are wrapped around one or multiple frameworks. The most wellknown and widely-used interface for deep learning today is Keras. Keras is a high-level deep learning API, written in Python.

Deep learning development flow

1. Selection of a framework for development
2. Selecting labeled data set of classes to train the network upon
3. Designing initial network model
4. Training the network
5. Saving the parameters and architecture in a binary file
6. Inference



Training data:



Introduction to deep learning summary

• What is deep learning? Artificial intelligence, or AI, is an umbrella term for any computer program that does something smart. Machine learning is a subset of AI and Deep Learning is subset of Machine Learning.

• Deep learning has its roots in neural networks.

• Neural networks are sets of algorithms, modeled loosely after the human brain, that are designed to recognize patterns.

• Speaking deep learning: Network types, nodes, layers, development frameworks and network models.

• Deep learning solution development flow

• Application spaces

CNN:

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivalent representations. Moreover, convolution provides a means for working with inputs of variable size. Traditional neural network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit. This means every output unit interacts with every input unit. Convolutional networks, however, typically have sparse interactions This is accomplished by making the kernel smaller than the input. For example, when processing an image, the input image might have thousands or millions of pixels, but we can detect small, meaningful features such as edges with kernels that occupy only tens or hundreds of pixels.

Store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency. It also means that computing the output requires fewer operations. These improvements in efficiency are usually quite large For many practical applications, it is possible to obtain good performance on the machine learning task while keeping several orders of magnitude less data. This allows the network to efficiently describe complicated interactions between many variables by constructing such interactions from simple building blocks that each describe only sparse interactions.

The Future of Deep Learning

The performance of deep learning systems can often be dramatically improved by simply scaling them up. With a lot more data and a lot more computation, they generally work a lot better. The language model GPT-318 with 175 billion parameters (which is still tiny compared with the number of synapses in the human brain) generates noticeably better text than GPT-2 with only 1.5 billion parameters. The chatbots Meena and BlenderBot also keep improving as they get bigger. Enormous effort is now going into scaling up and it will improve existing systems a lot, but there are fundamental deficiencies of current deep learning that cannot be overcome by scaling alone, as discussed here.

Comparing human learning abilities with current AI suggests several directions for improvement:

1. Supervised learning requires too much labeled data and model-free reinforcement learning requires far too many trials. Humans seem to be able to generalize well with far less experience.

2. Current systems are not as robust to changes in distribution as humans, who can quickly adapt to such changes with very few examples.

3. Current deep learning is most successful at perception tasks and generally what are called system 1 tasks. Using deep learning for system 2 tasks that require a deliberate sequence of steps is an exciting area that is still in its infancy. What needs to be improved. From the early days, theoreticians of machine learning have focused on the id assumption, which states that the test cases are expected to come from the same distribution as the training examples. Unfortunately, this is not a realistic assumption in the real world: just consider the non-stationarities due to actions of various agents changing the world, or the gradually expanding mental horizon of a learning agent which always has more to learn and discover. As a practical consequence, the performance of today's best AI systems tends to take a hit when they go from the lab to the field.

Our desire to achieve greater robustness when confronted with changes in distribution (called out-of-distribution generalization) is a special case of the more general objective of reducing sample complexity (the number of examples needed to generalize well) when faced with a new task—as in transfer learning and lifelong learning or simply with a change in distribution or in the relationship between states of the world and rewards. Current supervised learning systems require many more examples than humans (when having to learn a new task) and the situation is even worse for model-free reinforcement learning since each rewarded trial provides less information about the task than each labeled example. It has already been noted that humans can generalize in a way that is different and more powerful than ordinary id generalization: we can correctly interpret novel combinations of existing concepts, even if those combinations are extremely unlikely under our training distribution, so long as they respect high-level syntactic and semantic patterns we have already learned. Recent studies help us clarify how different neural net architectures fare in terms of this systematic generalization ability.

Introduction to Python:

Python language was the successor of ABC language. It was created by Guido Van Rossum in the year 1991. It is an interpreted, general purpose and high level programming language which emphasizes on the readability of code with its significant whitespace. In this an object oriented approach was followed in order to support multiple programming paradigms. Python is basically dynamically typed and also garbage collected that includes the procedural and functional programming. Next versions of python were:

- In the year 2000, the next version python 2.0 is released.

- Later in the year 2008, python 3.0 was released.

The formatting in the python was uncluttered visually and more oftenly it uses the English keywords. The program semantic structure is generally represented by programs visual structure. This language does not use the curly brackets and semicolons but instead uses the whitespace indentation to delimit the blocks. The following are specifications of indentation for constructing the blocks:

- An increase in indentation will come after certainstatements

- A decrease in indentation specifies the end of the presentblock.

1. Statements:

- For assigning a value to a variable, there is no need for specifying the data-type. The successive assignments of a common value to multiple variables can result in allocating the storage to names andobject.
- The if statement will definitely execute along with the else block and also elif (i.e., else if) block.
- “import” statements are generally used for import modules whose variables/functions can be used in the present program. It can specified by using anyone of 3ways:

i. import <module> as [<formalname>]

ii. from <module> import*

iii. from <module> import <function1> [<formal name1>], <function2> [<formalname2>] 2Expressions:

- The arithmetic operations performs same as like as other programming language but the only change is occurred during the division. In python there are 2 types ofdivisions:

i. Floor division (//)

ii. Floating point division(/)

- From python 3.5 version, an infix operator that is denoted by @ is being used by the libraries of the NumPy for matrixmultiplication.
- In python, + is used for concatenation of tuples and % for stringformat.

Note:

i. The operator that is being used for exponentiation is**.

ii. Tuples are immutable, can be denoted by ( ) and hence these can be used as keys of dictionaries.

iii. Lists are mutable, can be denoted by [ ] and hence these cannot be used as keys of dictionaries.

- Slice(:) will return a copy of entire tuple, list and each element is referred as a shallow copy. Slice will take elements from start index but does not include the stopindex.

3. Libraries:

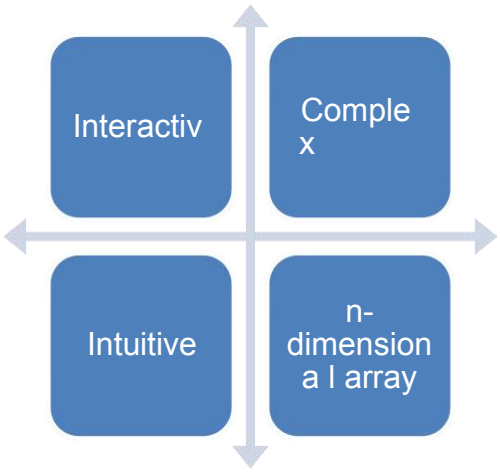
The most popular 10 libraries which are being used by the developers for the implementation of machine learning in the existing applications are:

- TensorFlow
- Pandas
- Scikit-Learn
- NumPy
- Keras
- Pytorch
- LightGBM

- Eli5
- SciPy
- Theano

1. NumPy:

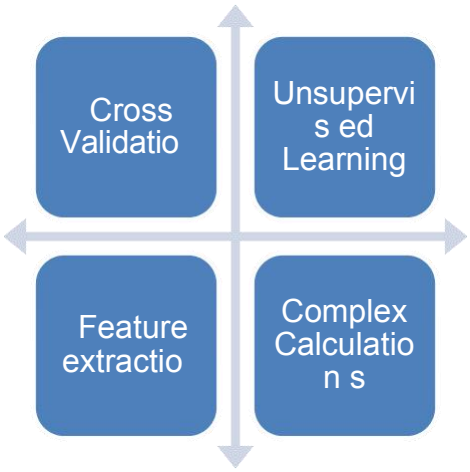
The abbreviation is numerical python, it contains the basic linear algebra, Fourier transformations, etc. The main feature is Array interface, which can be utilized for expressing the images and binary row streams as an array containing real numbers. The features are:



Numpy Features

2. Scikit-Learn:

Generally it is a python library associated with both Numpy and SciPy that works with complex data. This tool is mainly used for classification, regression, clustering and dimensionality reduction. The name is given as “sklearn” that is used for importing the modules from it. The features are:

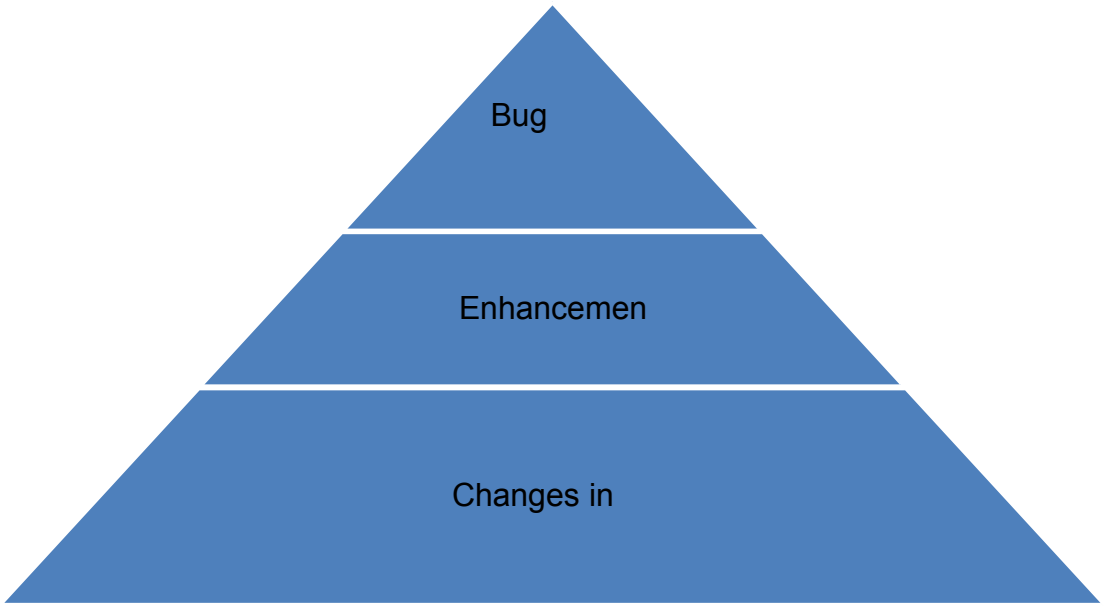


Scikit Features

3. **Pandas:**

This library mainly provides the data structures of high level and variety of tools for the analyzing. It performs the structured data operations and manipulations. The operations that are performed:

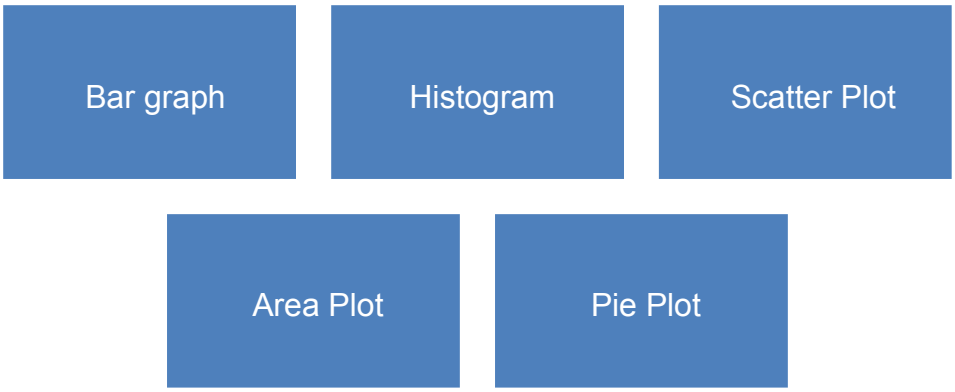
Slicing, Data Munging, Concatenation, Changing the index, Changing the column headers. The features are:



Pandas Features

4. **Matplotlib:**

This is a plotting library that is used for 2D graphics in python programming language. It is even used in python scripts, shell, web application servers and other GUI toolkits. The main disadvantages are this module is heavily reliant on other packages and only works for python but not any other high level programming languages. The types of plots are:

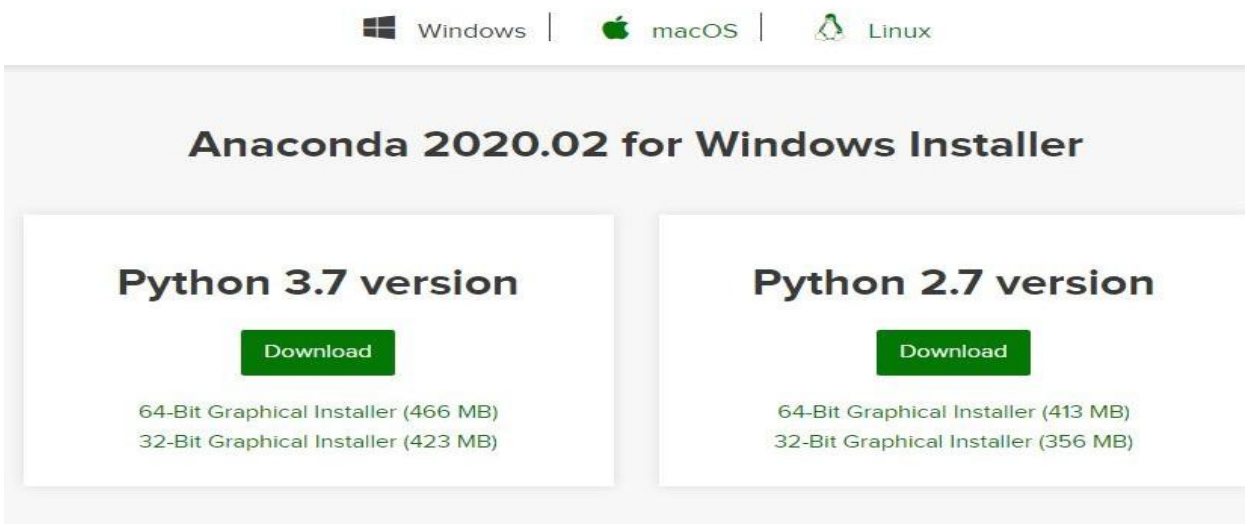


Types of Matplotlib

ANACONDA(A PYTHONDISTRIBUTION):

Installation of Anaconda:

Step 1: Go to anaconda navigator website for downloading.

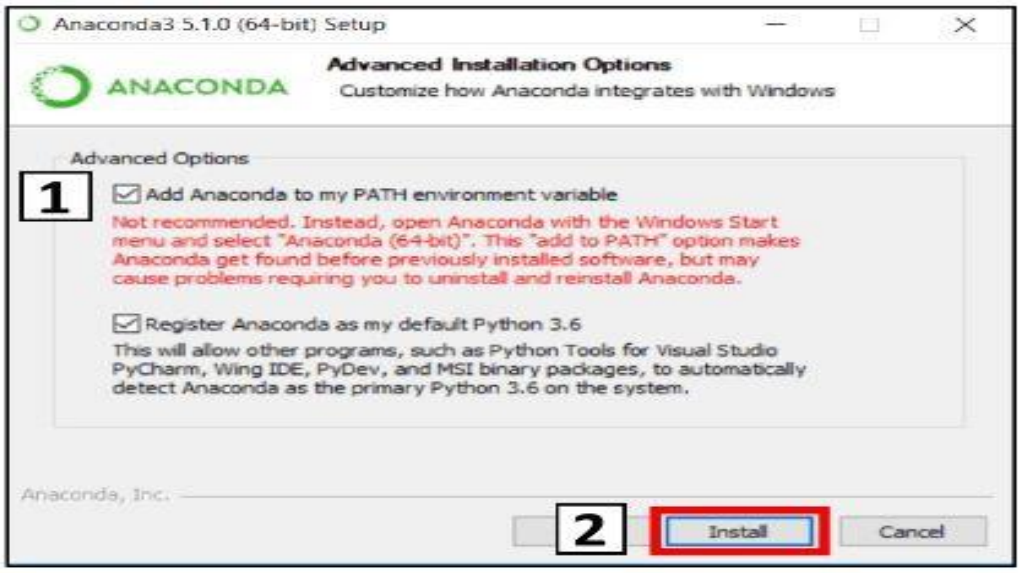


https://www.anaconda.com/distribution/

Anaconda navigator installer

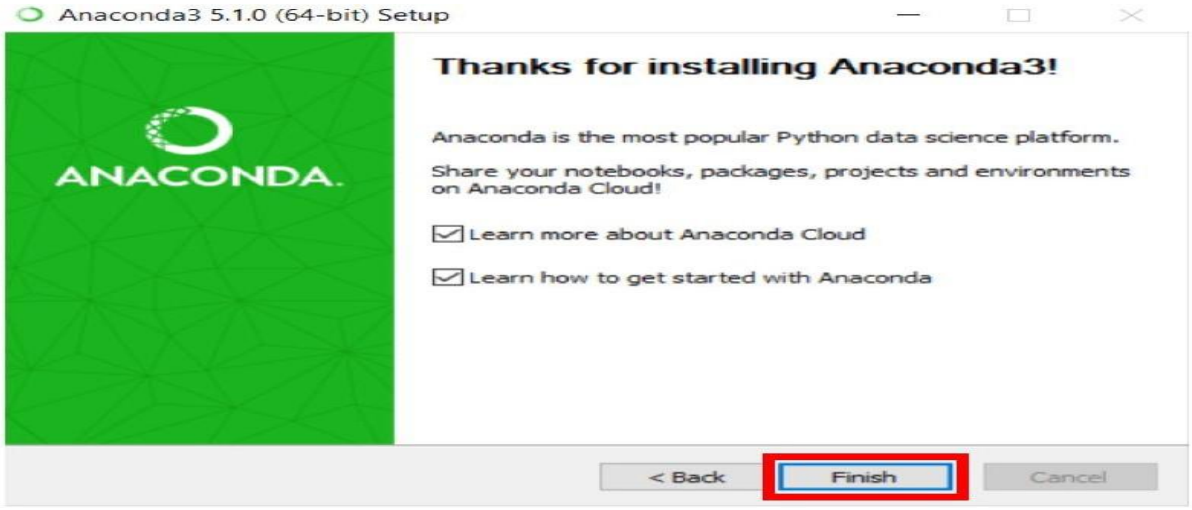
Step 2: Click on Anaconda3 that is available in Downloads.

Step 3: Click on Next and then accept the terms and conditions. After which a popup will come to select the path and the options will displayed for installation.



Installation options

Step 4: Click on install button and hence the following screens will be displayed.



Installation Successful

Introduction:

This is a free and open source distribution for R and Python programming languages for computing the scientific calculations that aims for the simplification of package management and deployment. The “conda” which is a package management system is responsible for the managing the versions. This distribution includes the data science packages, which are suitable for MacOS, Linux and Windows, and hence it comes up with 1500 packages that were selected from PyPI, virtual environment manager and conda. Basically it has 2 types ofmanagers:

i. pip (preferred installerprogram):

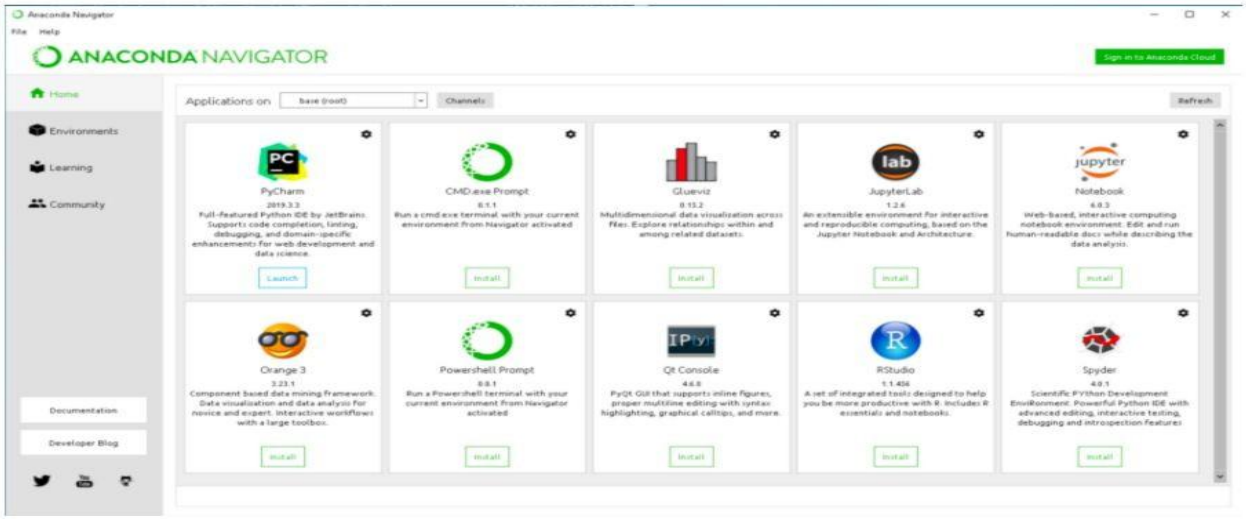
When a package is being installed with pip, then it automatically installs any dependent python package without noticing the previous installed packages.

ii. conda:

This manager will carefully analyze the present environment that includes the current and previously installed packages, then installs the missing one. The following command is used for installation: **conda install.**

The available packages on PyPI can be installed into the environment of conda using “pip” and the conda will maintain a track on what is being installed. The default installation will include python 3.7.

A Desktop GUI (graphical user interface) which allows to launch the applications and also maintains the packages, environments without any command line commands is Anaconda navigator.



Anaconda Home page

6.SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that 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.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

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 integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its

purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test Cases

USER REQUIREMENTS:

1. Home

Use case ID	Eye Disease Detection using Machine Learning
Use case Name	Home button
Description	Display home page of application
Primary actor	User
Precondition	User must open application
Post condition	Display the Home Page of an application
Frequency of Use case	Many times
Alternative use case	N/A
Use case Diagrams	
Attachments	N/A

7.SCREENSHOTS

8.CONCLUSION

The various computer technologies have helped the automation of detecting the OD accurately. A large number of researchers have reported on improved specificity and sensitivity in diagnosing glaucoma. A survey of those technologies [3], [4], [5] has also been reported by various authors in analyzingthe retinal image features. Machine intelligence technologies such as ANNs have helped the physicians in management and analysis of clinical data. When a neural network model is employed for medical image processing, compared with conventional image processing methods, there are several things that are to beconsidered. Firstly, the time for applying a trained neural network to solve a medical imageprocessing problem is negligibly small, though the training of a neural network is a time consuming work. Secondly, medical image processing tasks often require complex computations. Hence the performance of ANNs can be significantly improved when image based information is added, which shows the need for (semi-)automatic segmentation and modelling.

9. REFERENCES

[1]. Acharya, U. R., Kannathal, N., Ng, E. Y. K., Min, L. C., & Suri, J. S. (2006, August), Computer-based classification of eye diseasesIn Engineering in

Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE (pp. 6121-6124). IEEE.

[2]. Gardner, G. G., Keating, D., Williamson, T. H., & Elliott, A. T. (1996), Automatic detection of diabetic retinopathy using an artificial neural network: a screening toolBritish journal of Ophthalmology, 80(11), 940- 944.

[3]. Winder, R. J., Morrow, P. J., McRitchie, I. N., Bailie, J. R., & Hart, P. M. (2009), Algorithms for digital image processing in diabetic retinopathyComputerized medical imaging and graphics, 33(8), 608-622.

[4]. Ibrahim, F., Ali, J. B., Jaais, A. F., & Taib, M. N. (2001), Expert system for early diagnosis of eye diseases infecting the Malaysian population In TENCON 2001. Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology (Vol. 1, pp. 430-432). IEEE.

[5]. Bock, R., Meier, J., Michelson, G., Nyúl, L. G., & Hornegger, J. (2007, September), Classifying glaucoma with image-based features from fundus photographsIn Joint Pattern Recognition Symposium (pp. 355-364). Springer, Berlin, Heidelberg.

[6]. Sajda, P. (2006), Machine learning for detection and diagnosis of disease Annu. Rev. Biomed. Eng., 8, 537-565.

[7] Jelinek, H. F., Rocha, A., Carvalho, T., Goldenstein, S., & Wainer, J. (2011, August)Machine learning and pattern classification in identification of indigenous retinal pathologyIn Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE (pp. 5951-5954). IEEE.