A Project/Dissertation Review-1
Report
on
BRAIN TUMOR CLASSIFICATION USING MOBILENET

Submitted in partial fulfillment of the

requirement for the award of the degree

of

B.TECH CSE

Under The Supervisionof
Mr. Deependra Rastogi
Assistant Professor

Submitted By

Shubham Kumar

19SCSE1010369

Akshay Pratap

19SCSE1010533

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER,2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"CAPS…."** in partial fulfillment of the requirements for the award of the B. tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Mr. Deependra Rastogi Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Shubham Kumar, 19SCSE1010369

Akshay Pratap, 19SCSE1010533

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Deependra Rastogi

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Shubham Kumar:19SCSE1010369 & Akshay Pratap:19SCSE1010533 has been held on_____and his/her work is recommended for the award of B.tech.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2013

Place: Greater Noida

Abstract

The brain tumors, are the most common and aggressive disease, leading to a very short life expectancy in their highest grade. Generally, various image techniques such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and ultrasound image are used to evaluate the tumor in a brain. Earlier reviewing is done manually. However, manually reviewing these images is time-consuming, hectic, and even prone to error due to the influx of patients.

We propose a Convolutional Neural Network (CNN) approach which is amongst the top performing methods while also being extremely computationally efficient, Our CNN is trained directly on the image modalities and thus learns a feature representation directly from the data. Our automatic hybrid scheme for brain tumor classification, uses both (1) the pre-trained CNN models to extract the deep features from brain MR images and (2) ML classifiers to classify brain tumor type effectively. It consists of three steps: (1) extract deep features using pre-trained CNN models, (2) select the top three performing features, (3) combine them to build the ensemble model.

In our project technology and tools which is used are Machine Learning (ML) techniques, MobileNet (i.e Convolutional Neural Network),Magnetic Resonance Imaging (MRI),ML classifiers.

We conducted extensive experiments on 13 different pre-trained CNN models and 9 different ML classifiers to compare the effectiveness of each pre-trained CNN model and each ML classifier on three different brain MRI datasets . The top three deep features are concatenated in our ensemble module, and the concatenated deep features are further used as an input to ML classifiers to predict final output and classify the MRI imaging.
In summary, we presented a brain tumor classification method using the ensemble of deep features from pre-trained MobileNet networks with ML classifiers we used pre- trained deep convolutional neural network to extract deep features from brain MR images

Table of Contents

List of Figures

CHAPTER-1
Introduction

1. INTRODUCTION

In the human body, the brain is an enormous and complex organ, and it contains around 100-billion nerve cells. This essential organ is originated in the center of the nervous system, Therefore, any kind of abnormality that exists in the brain may put human health in danger. Among such abnormalities, brain tumors are the most severe ones. According to the WHO, brain tumors can be classified into four grades. The grade1,grade2 tumors describe lower-level tumors(e.g., meningioma), while grade3,grade4 tumors consist of more severe ones(e.g., glioma). In clinical practice, the incidence rates of meningioma, pituitary, and glioma tumors are approximately 15%, 15%, and 45%, respectively.

To address this problem, the development of an automatic computer-aided diagnosis (CAD) system is required to alleviate the workload of the classification and diagnosis of brain MRI and act as a tool for helping radiologists and doctors. we proposed a hybrid solution that exploits various pre-trained deep convolutional neural networks (CNNs) to extract deep features from brain magnetic resonance (MR) images, and various ML classifiers to identify the normal and abnormal brain MR images. In our project we are using MobileNet CNN MODEL to help classifier to identify the normal and abnormal brain MR images.

1.2 FORMULATION OF PROBLEM

The main problem is that Several efforts have been made to develop a highly accurate and robust solution for the automatic classification of brain tumors. However, due to high inter and intra shape, texture, and contrast variations, it remains a challenging problem. The traditional machine learning (ML) techniques rely on handcrafted features, which restrains the robustness of the solution, manually reviewing the MRI images is time-consuming, hectic, and even prone to error due to the influx of patients. Whereas the deep learning-based techniques automatically extract meaningful features which offer significantly better performance. However, deep learning-based techniques require a large amount of annotated data for training, and acquiring such data is a challenging task.

1.2.1 Tools and Technology Used

The tools and technology that we will use for the proposed system will be as follows:-

- We will use python language for the proposed system of BRAIN TUMOR CLASSIFICATION.
- We will use the technology of machine learning for this proposed project.
- For this proposed project we would require the deep learning of machine learning algorithms.
- Mainly here supervised learning is used.
- Biometric analysis will be used to detect the different BRAIN TUMOR CLASSIFICATION.
- In clustering techniques, K means clustering algorithm, FCM clustering algorithm and expectation maximization (EM) algorithm are most widely used brain tumor detection techniques.

The first step in developing anything is to state the requirements. This applies just as much to leading edge research as to simple programs and to personal programs, as well as to large team efforts. Being vague about your objective only postpones decisions to a later stage where changes are much more costly.

The problem statement should state what is to be done and not how it is to be done. It should be a statement of needs, not a proposal for a solution. A user manual for the desired system is a good problem statement. The requestor should indicate which features are mandatory and which are optional, to avoid overly constraining design decisions. The requestor should avoid describing system internals, as this restricts implementation flexibility. Performance specifications and protocols for interaction with external systems are legitimate requirements. Software engineering standards, such as modular construction, design for testability, and provision for future extensions, are also proper.

Many problems statements, from individuals, companies, and government agencies, mixture requirements with design decisions. There may sometimes be a compelling reason to require a particular computer or language; there is rarely justification to specify the use of a particular algorithm. The analyst must separate the true requirements from design and implementation decisions disguised as requirements. The analyst should challenge such pseudo requirements, as they restrict flexibility. There may be politics or organizational reasons for the prequirements, but at least the analyst should recognize that these externally imposed design decisions are not essential features of the problem domain.

A problem statement may have more or less detail. A requirement for a conventional product, such as a payroll program or a billing system, may have considerable detail. A requirement for a research effort in a new area may lack many details, but presumably the research has some objective, which should be clearly stated.

Most problem statements are ambiguous, incomplete, or even inconsistent.

Some requirements are just plain wrong. Some requirements, although precisely stated, have unpleasant consequences on the system behavior or impose unreasonable implementation costs. Some requirements seem reasonable at first but do not work out as well as the request or thought. The problem statement is just a starting point for understanding the problem, not an immutable document. The purpose of the subsequent analysis is to fully understand the problem and its implications. There is no reasons to expect that a problem statement prepared without a fully analysis will be correct.

The analyst must work with the requestor to refine the requirements so they represent the requestor's true intent. This involves challenging the requirements and probing for missing information. The psychological, organizational, and political considerations of doing this are beyond the scope of this book, except for the following piece of advice: If you do exactly what the customer asked for, but the result does not meet the customer's real needs, you will probably be blamed.

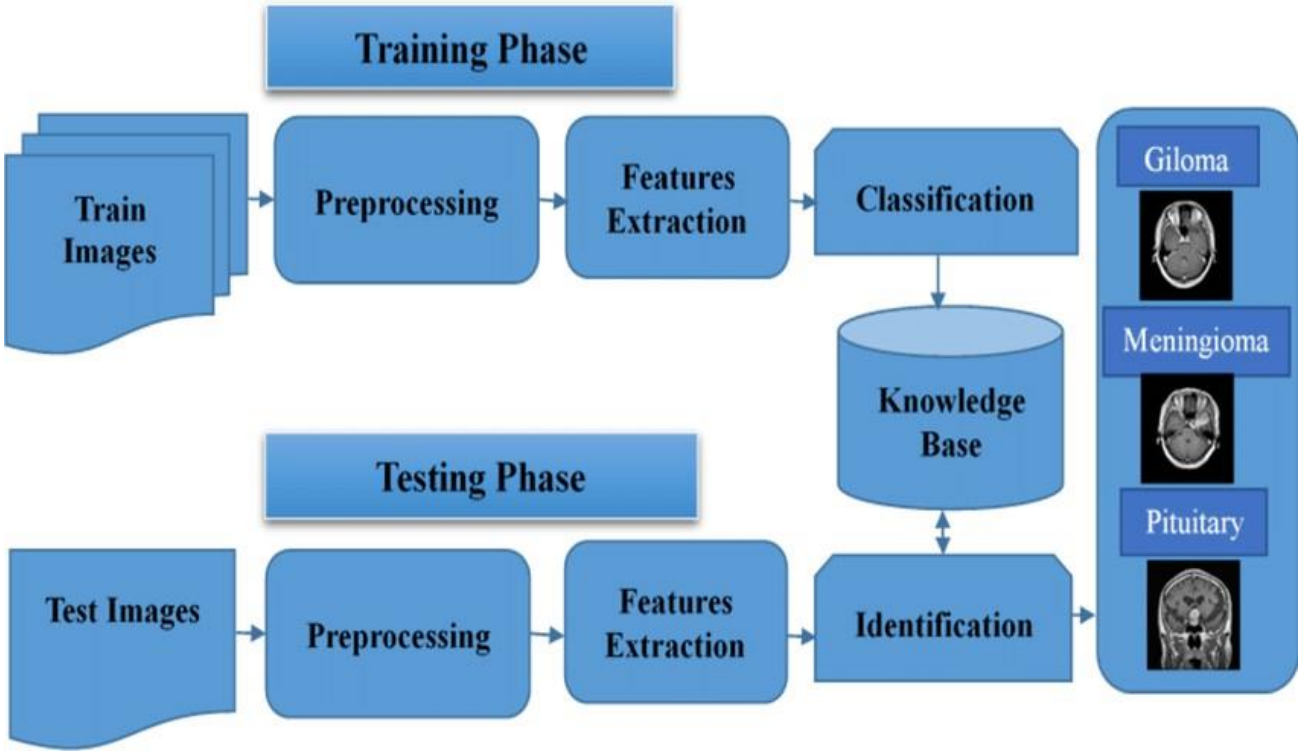
CHAPTER-2
Literature Survey

Numerous techniques have been proposed for automatic brain MRI classification based on traditional ML and deep learning methods The traditional ML methods are comprised of several steps: pre-processing, feature extraction, feature reduction, and classification. In traditional ML methods, feature extraction is a core step as the classification accuracy relies on extracted features. There are two main types of feature extraction. The first type of feature extraction is low-level (global) features, for instance, texture features and intensity, first-order statistics (e.g., mean, standard deviation, and skewness), and second-order statistics such as gray-level co- occurrence matrix (GLCM), wavelet transform (WT), The second type of feature extraction is the high-level (local) features, such as fisher vector (FV), scale- invariant feature transformation (SIFT), and bag-of-words (BoW). Different researchers have employed BoW for medical image retrieval and classification. Such as the classification of breast tissue density in mammograms [11], X-ray images retrieval and classification on pathology and organ levels [12], and content-based retrieval of brain tumor [13]. Cheng et al. [14] employed FV to retrieve the brain tumor. Most of the existing works in medical MR imaging refers to automatic segmentation of tumor region. Recently, Numerous researchers have proposed different techniques to detect and segment the tumor region in MR images [15–17]. Once the tumor in MRI is segmented, these tumors need to be classified into different grades. In previous research studies, binary classifiers have been employed to identify the benign and malignant classes.

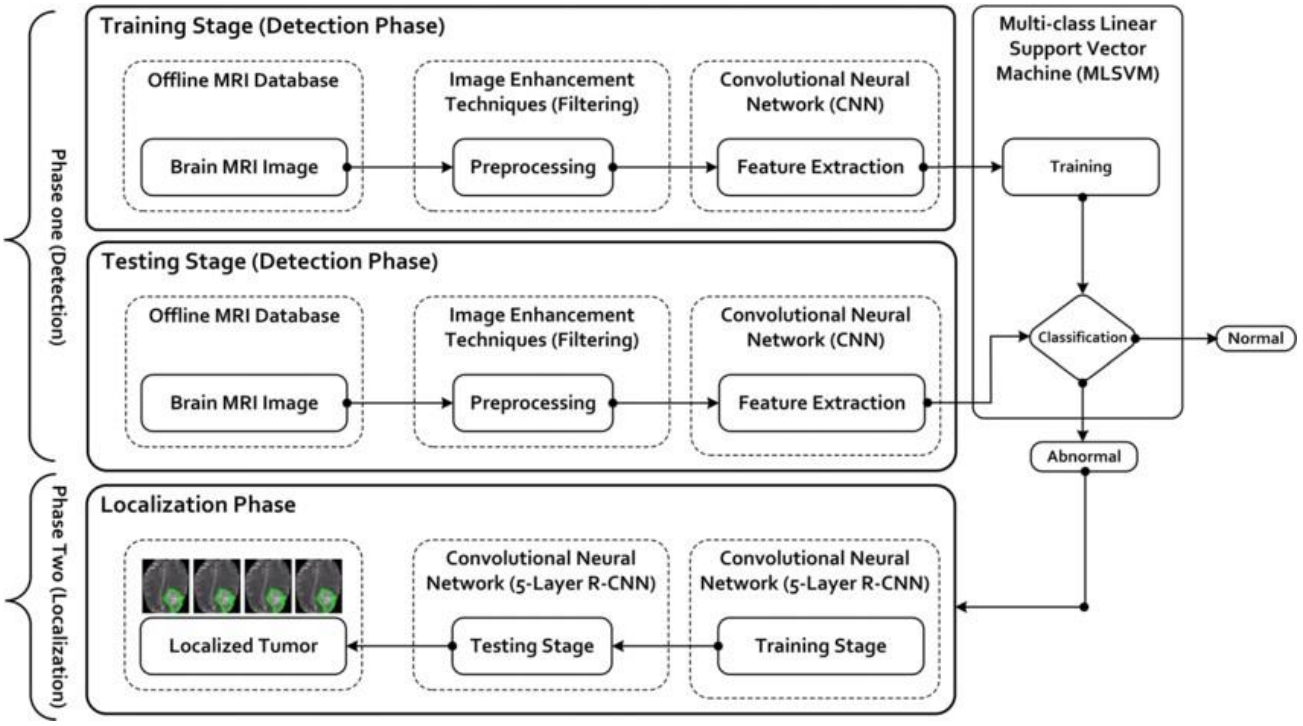
Since the last decade, deep learning methods have been widely used for brain MRI classification [23,24]. The deep learning method does not need handcrafted (manually) Sensors 2021, 21, 2222 5 of 21 extracted features as it embedded the feature extraction and classification stage in selflearning. The deep learning method requires a dataset where sometimes a pre-processing operation needs to be done, and then salient features are determined in a self-learning manner [25]. In MR imaging classification, a key challenge is to reduce the semantic gap between the high-level visual information perceived by the human evaluator and the lowlevel visual information captured by the MR imaging machine. To reduce the semantic gap, the convolutional neural networks (CNNs), one of the famous deep learning techniques for image data, can be used as a feature extractor to capture the relevant features for the classification task.

CHAPTER 3
WORKING OF PROJECT

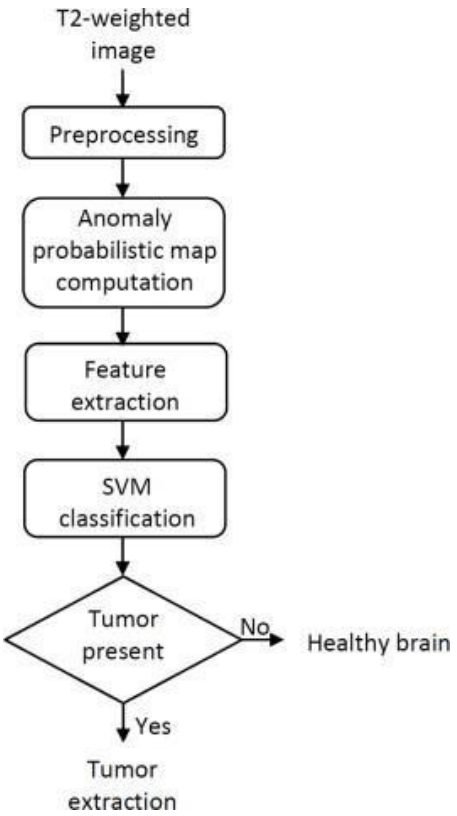
DATA FLOW DIAGRAM (FOR PROPOSED SYSTEM):-



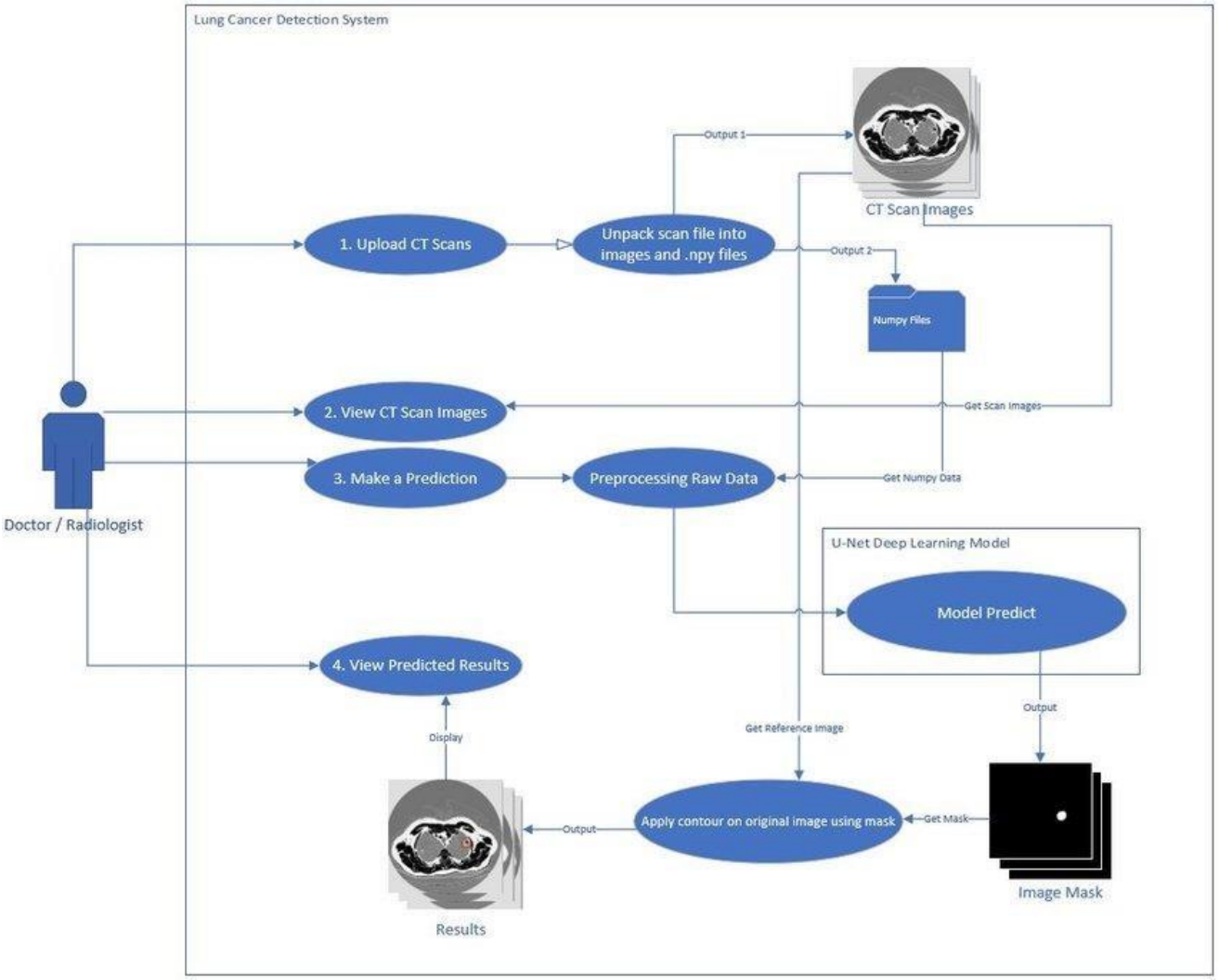
FLOW CHART (FOR PROPOSED SYSTEM):-

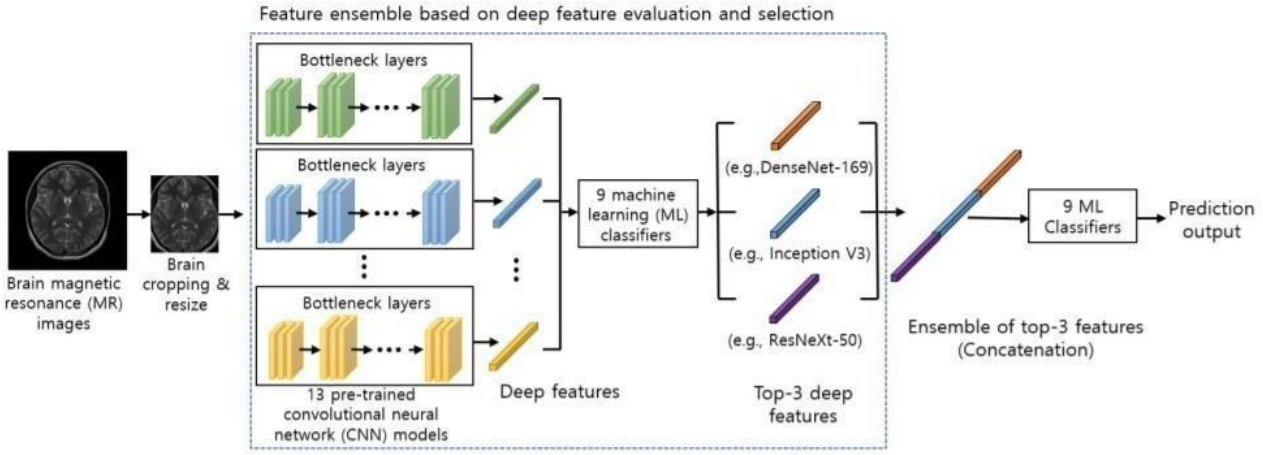


ACTIVITY DIAGRAM (FOR PROPOSED SYSTEM):-



USE CASE DIAGRAM (FOR PROPOSED SYSTEM):-





Architecture diagram

FEATURES OF LANGUAGE USED:-

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle- detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward- compatible. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages. Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with
the Amoeba operating system. Its implementation began in December 1989.Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member "Steering Council" to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features, including a cycle- detecting garbage collector (in addition to reference counting) for memory management and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major

features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the `2to3` utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported. Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

I have used the dataset available on kaggle

It has 198 images as training set and 58 images as test sets.

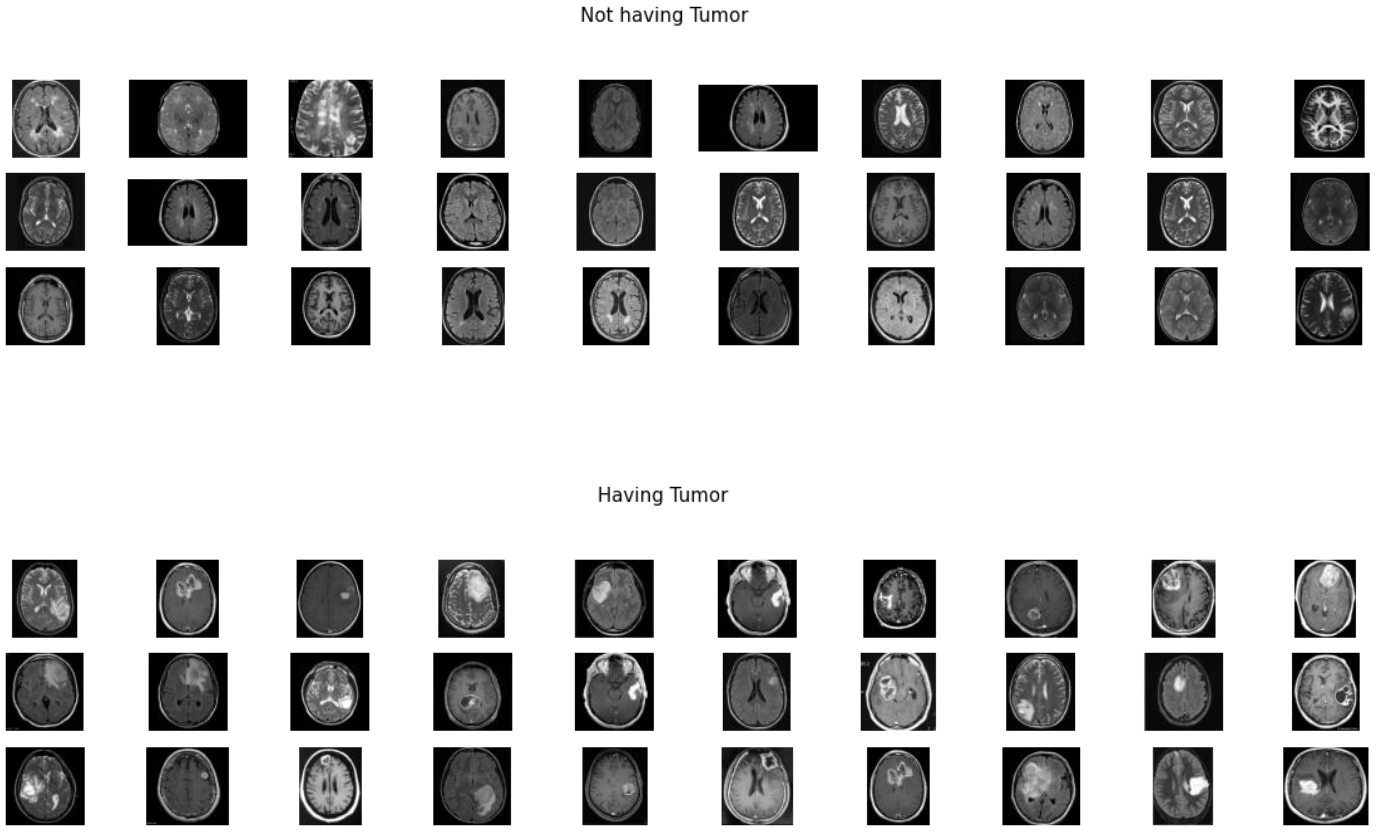
The dataset folder has been divided into training and test set folders which is further divided into yes or no folders.

Single prediction folder contains single images of brain scans that are used to validate whether the model can predict correctly or not.

convolution_neural_network.py contains the main CNN code for classifying the MRI scan,this model has validation accuracy .

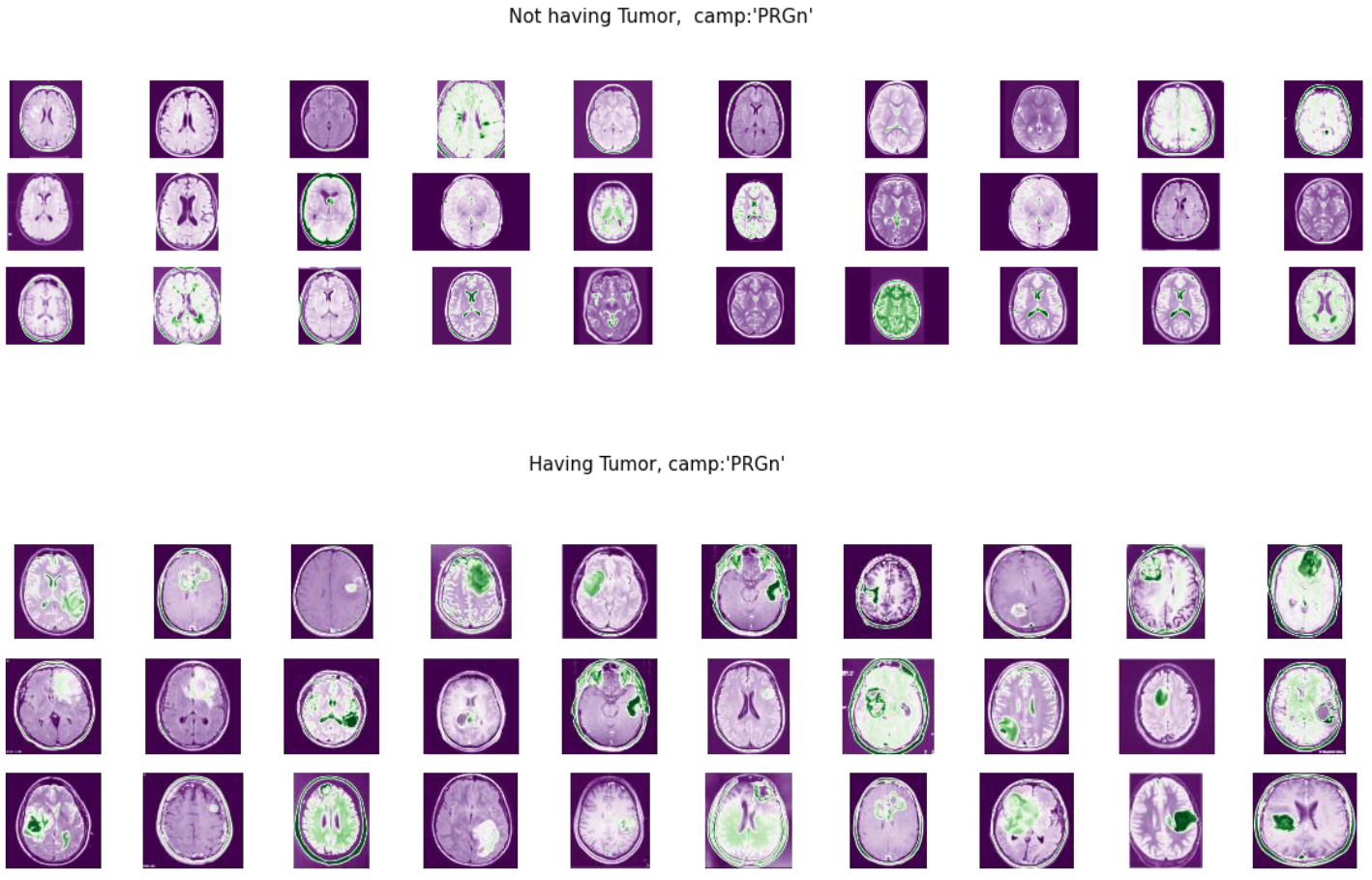
Visualizing with Colormaps

Colormaps control the way data are understood when plotted because they map data values to color. When the colors used do not match the way humans perceive color, there can be a mismatch between interpretation and the data.Map areas are colored, two-dimensional areas on a map that represent geographic regions, such as countries, states, and counties. ... You can color-code areas based on the value of a metric.

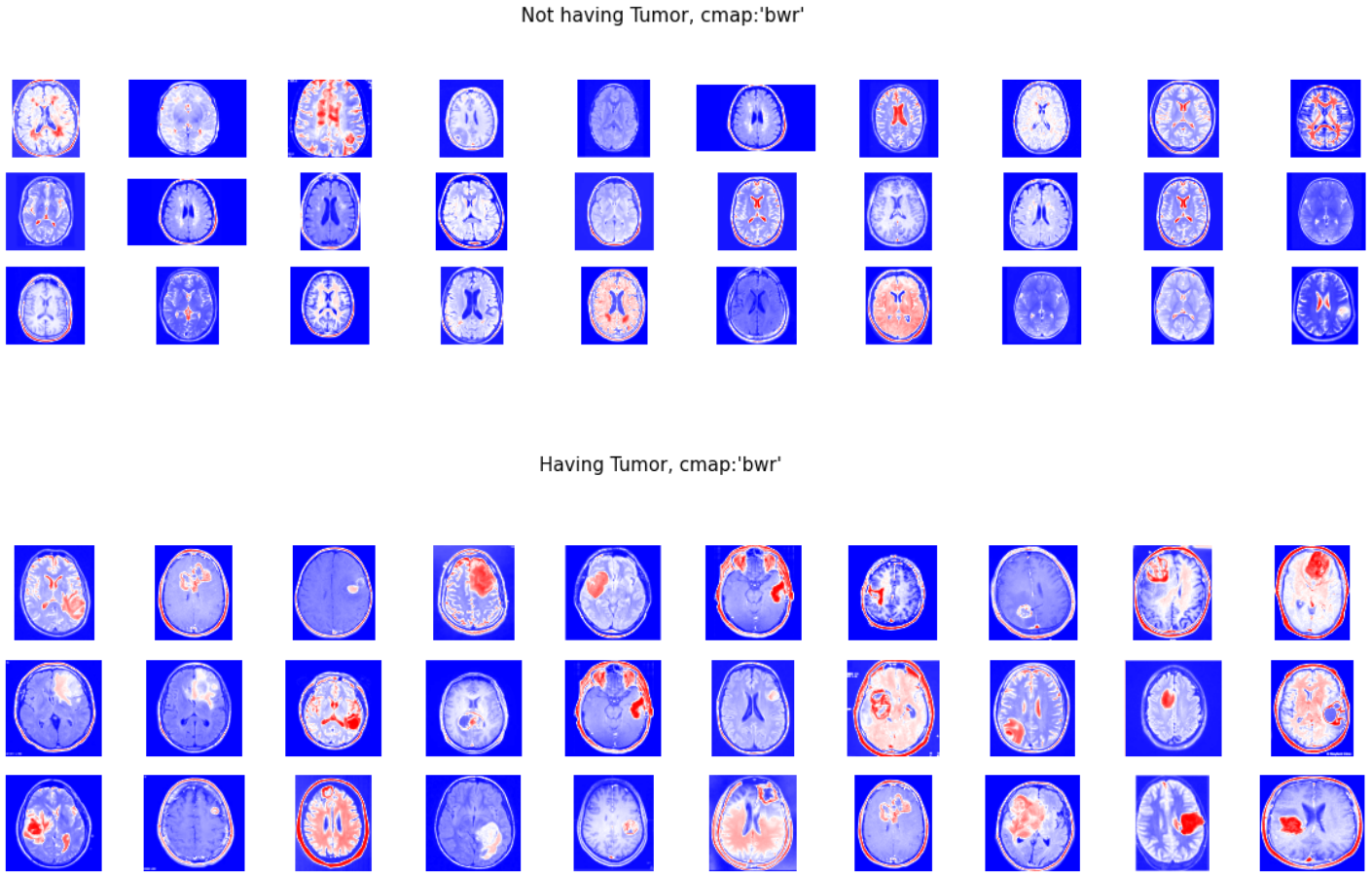


Random number generation

Random number generation is a process by which, often by means of a random number generator (RNG), a sequence of numbers or symbols that cannot be reasonably predicted better than by random chance is generated. This means that the particular outcome sequence will contain some patterns detectable in hindsight but unpredictable to foresight. True random number generators can be *hardware random-number generators* (HRNGS) that generate random numbers, wherein each generation is a function of the current value of a physical environment's attribute that is constantly changing in a manner that is practically impossible to model. This would be in contrast to so-called "random number generations" done by *pseudorandom number generators* (PRNGs) that generate numbers that only look random but are in fact pre-determined—these generations can be reproduced simply by knowing the state of the PRNG.



BWR



Equalized Hist

A histogram of an image is the graphical interpretation of the image's pixel intensity values. It can be interpreted as the data structure that stores the frequencies of all the pixel intensity levels in the image. Histogram Equalization is an image processing technique that adjusts the contrast of an image by using its histogram. To enhance the image's contrast, it spreads out the most frequent pixel intensity values or stretches out the intensity range of the image. By accomplishing this, histogram equalization allows the image's areas with lower contrast to gain a higher contrast. Histogram Equalization can be used when you have images that look washed out because they do not have sufficient contrast. In such photographs, the light and dark areas blend together creating a flatter image that lacks highlights and shadows

K-Means clustering

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.
K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

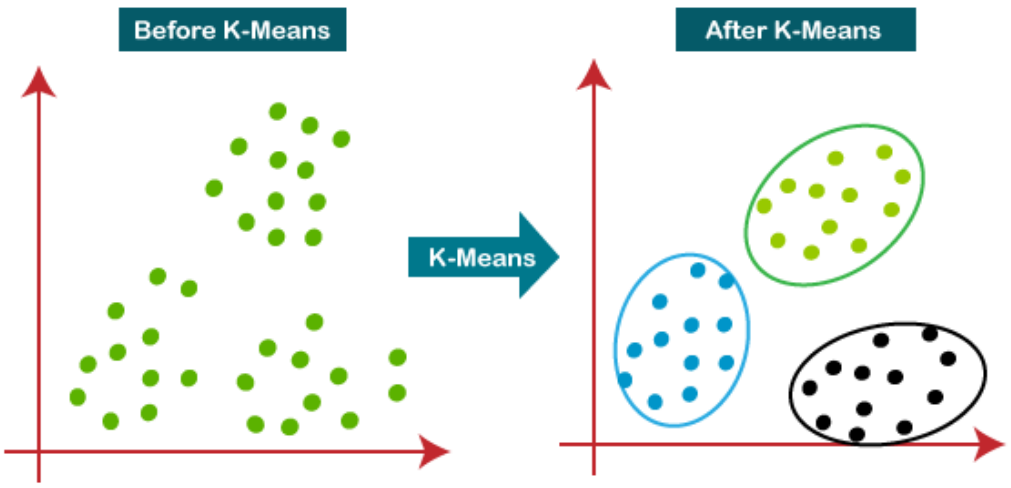
It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.



The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.
Step-2: Select random K points or centroids. (It can be other from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

Model traning with transfer learning:

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem.
In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest.
Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error.
The weights in re-used layers may be used as the starting point for the training process and adapted in response to the new problem. This usage treats transfer learning as a type of weight initialization scheme. This may be useful when the first related problem has a lot more labeled data than the problem of interest and the similarity in the structure of the problem may be useful in both contexts.

Statements and control flow

Python's statements include (among others):

- The assignment statement, using a single equals sign `=`.
- The `if` statement, which conditionally executes a block of code, along with `else` and `elif` (a contraction of else-if).
- The `for` statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The `while` statement, which executes a block of code as long as its condition is true.
- The `try` statement, which allows exceptions raised in its attached code block to be caught and handled by `except` clauses; it also ensures that clean-up code in a `finally` block will always be run regardless of how the block exits.
- The `raise` statement, used to raise a specified exception or re-raise a caught exception.
- The `class` statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The `def` statement, which defines a function or method.
- The `with` statement, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing resource-acquisition-is-initialization (RAII)-like behavior and replaces a common try/finally idiom.[81]
- The `break` statement, exits from a loop.
- The `continue` statement, skips this iteration and continues with the next item.
- The `del` statement, removes a variable, which means the reference from the name to the value is deleted and trying to use that variable will cause an error. A deleted variable can be reassigned.
- The `pass` statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The `assert` statement, used during debugging to check for conditions that should apply.
- The `yield` statement, which returns a value from a generator function and `yield` is also an operator. This form is used to implement coroutines.

The assignment statement (=) operates by binding a name as a reference to a separate, dynamically-allocated object. Variables may subsequently be rebound at any time to any object. In Python, a variable name is a generic reference holder and does not have a fixed data type associated with it. However, at a given time, a variable will refer to *some* object, which will have a type. This is referred to as dynamic typing and is contrasted with statically-typed programming languages, where each variable may only contain values of a certain type.

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will.[82][83] However, better support for coroutine-like functionality is provided, by extending Python's generators.[84] Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.[85]

EXPRESSION

Some Python expressions are similar to those found in languages such as C and Java, while some are not:

- Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division (or integer division) `//` and floating-point `/` division.[86] Python also uses the `**` operator for exponentiation.
- From Python 3.5, the new `@` infix operator was introduced. It is intended to be used by libraries such as NumPy for matrix multiplication.[87][88]
- From Python 3.8, the syntax `:=`, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.[89]
- In Python, `==` compares by value, versus Java, which compares numerics by value[90] and objects by reference.[91] (Value comparisons in Java on objects can be performed with the `equals()` method.) Python's `is` operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example `a <= b <= c`.
- Python uses the words `and`, `or`, `not` for its boolean operators rather than the symbolic `&&`, `||`, `!` used in Java and C.

- Python has a type of expression termed a *list comprehension* as well as a more general expression termed a *generator expression*.[64]
- Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.
- Conditional expressions in Python are written as `x` **if** `c` **else** `y`[92] (different in order of operands from the `c ? x : y` operator common to many other languages).

- Python makes a distinction between lists and tuples. Lists are written as `[1, 2, 3]`, are mutable, and cannot be used as the keys of dictionaries (dictionary keys must
  be immutable in Python). Tuples are written as `(1, 2, 3)`, are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable.

- The `+` operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided
- Python features *sequence unpacking* wherein multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in an identical manner to that forming tuple literals and, as a whole, are put on the left- hand side of the equal sign in an assignment statement. The statement expects
  an *iterable* object on the right-hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through and will iterate through it, assigning each of the produced values to the corresponding expression on the left.[94]

.[95]

SDLC METHDOLOGIES:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

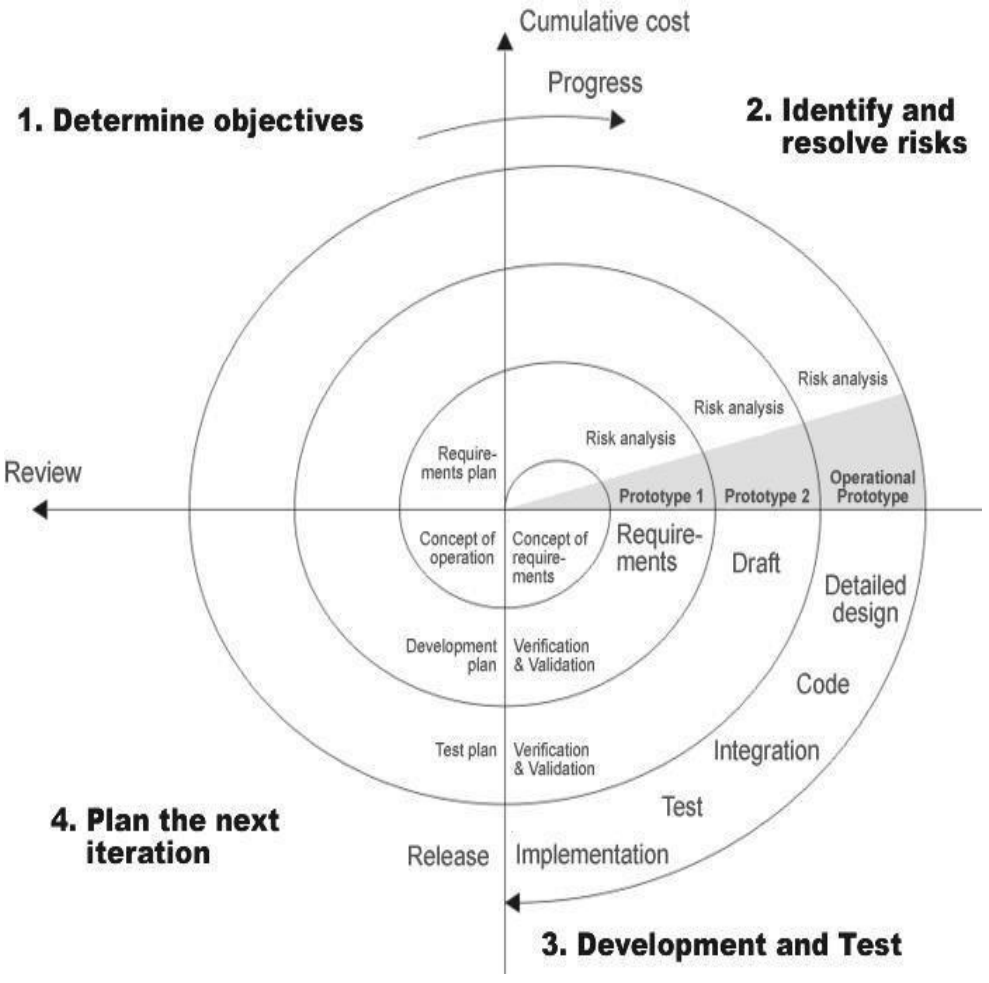
As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
-
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
- Evaluating the first prototype in terms of its strengths, weakness, and risks.
- Defining the requirements of the second prototype.
- Planning an designing the second prototype.

- Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.

- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

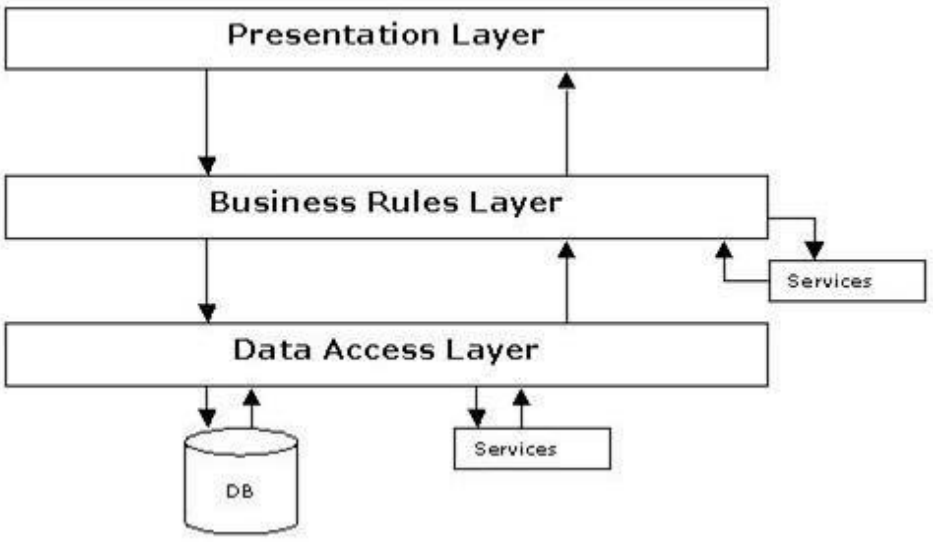


DEFINITION

Simply stated, an n-tier application helps us distribute the overall functionality into various tiers or layers:

- Presentation Layer
- Business Rules Layer
- Data Access Layer
- Database/Data Store

Each layer can be developed independently of the other provided that it adheres to the standards and communicates with the other layers as per the specifications.
This is the one of the biggest advantages of the n-tier application. Each layer can potentially treat the other layer as a ‘Block-Box’.
In other words, each layer does not care how other layer processes the data as long as it sends the right data in a correct format.



1.THE PRESENTATION LAYER

Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

2.THE BUSINESS RULES LAYER

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process.

3.THE DATA ACCESS LAYER

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

4.THE DATABASE LAYER

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc.
In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

MACHINE LEARNING

Machine learning (**ML**) is the study of computer algorithms that can improve automatically through experience and by the use of data.[1] It is seen as a part of artificial intelligence.

Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.[2] Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.[3]

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning.[5][6] Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain.[7][8] In its application across business problems, machine learning is also referred to as predictive analytics.

Association rules

Association rule learning is a rule-based machine learning method for discovering relationships between variables in large databases. It is intended to identify strong rules discovered in databases using some measure of "interestingness".[60]

Rule-based machine learning is a general term for any machine learning method that identifies, learns, or evolves "rules" to store, manipulate or apply knowledge. The defining characteristic of a rule-based machine learning algorithm is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system. This is in contrast to other machine learning algorithms that commonly identify a singular model that can be universally applied to any instance in order to make a prediction.[61]

Rule-based machine learning approaches include learning classifier systems, association rule learning, and artificial immune systems.

Inductive logic programming (ILP) is an approach to rule-learning using logic programming as a uniform representation for input examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesized logic program that entails all positive and no negative examples. Inductive programming is a related field that considers any kind of programming language for representing hypotheses (and not only logic programming), such as functional programs.

Artificial neural networks

An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non- linear function of the sum of its inputs. The connections between artificial neurons are called "edges". Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of

transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social
network filtering, playing board and video games and medical diagnosis

Decision tree

Decision tree learning uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining, and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making

Support-vector machines

Support-vector machines (SVMs), also known as support-vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.[69] An SVM training

algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high- dimensional feature spaces

Regression analysis

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares. The latter is often extended
by regularization (mathematics) methods to mitigate overfitting and bias, as in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression (for example, used for trendline fitting in Microsoft Excel[70]), logistic regression (often used
in statistical classification) or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher-dimensional space.

Training models

Usually, machine learning models require a lot of data in order for them to perform well. Usually, when training a machine learning model, one needs to collect a large, representative sample of data from a training set. Data from the training set can be as varied as a corpus of text, a collection of images, and data collected from individual users of a service. Overfitting is something to watch out for when training a machine learning model. Trained models derived from biased data can result in skewed or undesired predictions. Algorithmic bias is a potential result from data not fully prepared for training something to watch out for when training a machine learning model. Trained models derived from biased data can result in skewed or undesired predictions. Algorithmic bias is a potential result

from data not fully prepared for training.something to watch out for when training a machine learning model. Trained models derived from biased data can result in skewed or undesired predictions. Algorithmic bias is a potential result from data not fully prepared for training

Artificial intelligence

Part of machine learning as subfield of AI or part of AI as subfield of machine learning[22]

As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. In the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what was then termed "neural networks"; these were
mostly perceptrons and other models that were later found to be reinventions of the generalized linear models of statistics.[23] Probabilistic reasoning was also employed, especially in automated medical diagnosis.

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation.[24]: 488 By 1980, expert systems had come to dominate AI, and statistics was out of favor.[25] Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval.[24]: 708–710, 755 Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines.

USE OF SUPERVISED MACHINE LEARNING ALGORITHMS:-

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

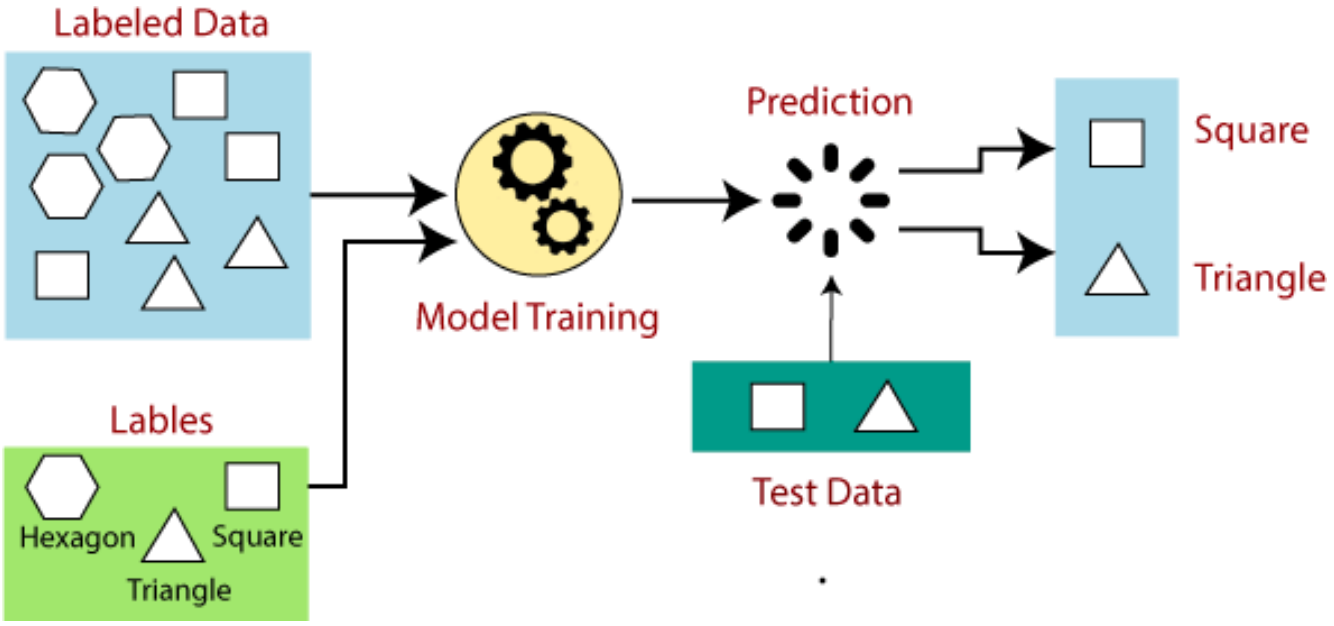
In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc

How Supervised Learning Works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output

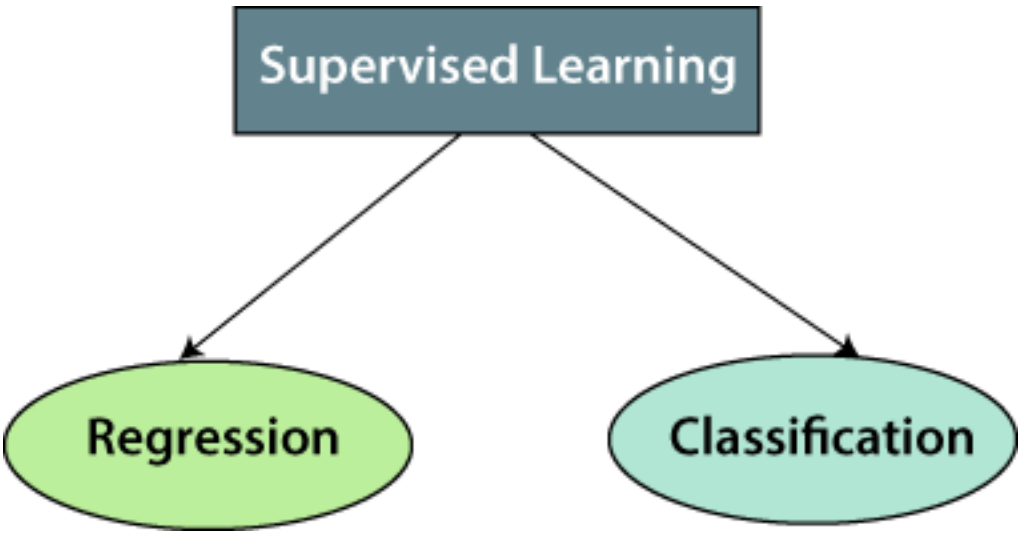


Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training **dataset, test dataset, and validation dataset**.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms

Supervised learning can be further divided into two types of problems:



1. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.
Spam Filtering,

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.

  Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**,

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.

  In supervised learning, we need enough knowledge about the classes of object

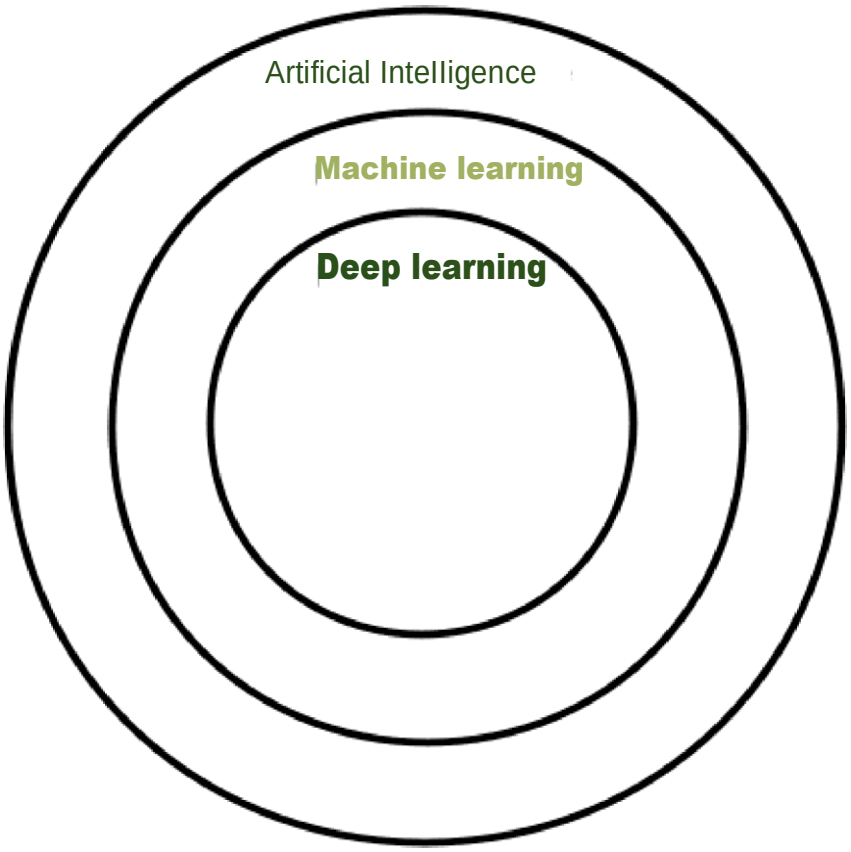
Deep learning:

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.

In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousand of their neighbours.
The question here is how do we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.

Architecture:

1. **Deep Neural Network** – It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.
2. **Deep Belief Network(DBN)** – It is a class of Deep Neural Network. It is multi-layer belief networks.
   Steps for performing DBN :
   a. Learn a layer of features from visible units using Contrastive Divergence algorithm.
   b. Treat activations of previously trained features as visible units and then learn features of features.
   c. Finally, the whole DBN is trained when the learning for the final hidden layer is achieved.
3. **Recurrent** (perform same task for every element of a sequence) **Neural Network** – Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.

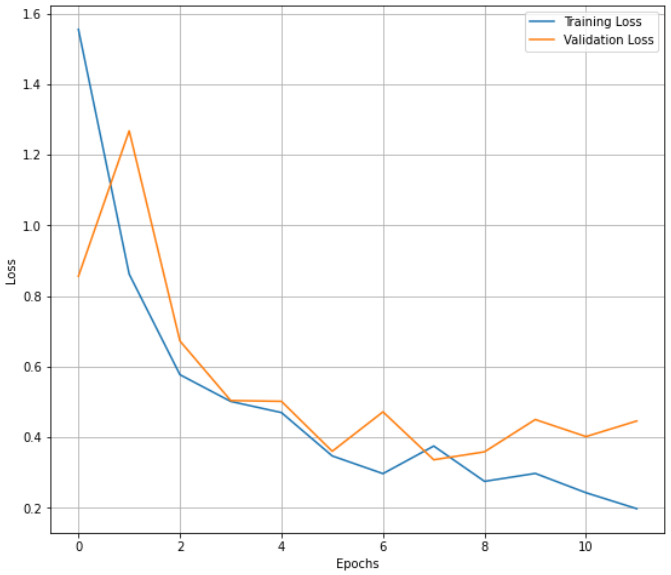

Artificial Intelligence
Machine learning
Deep learning

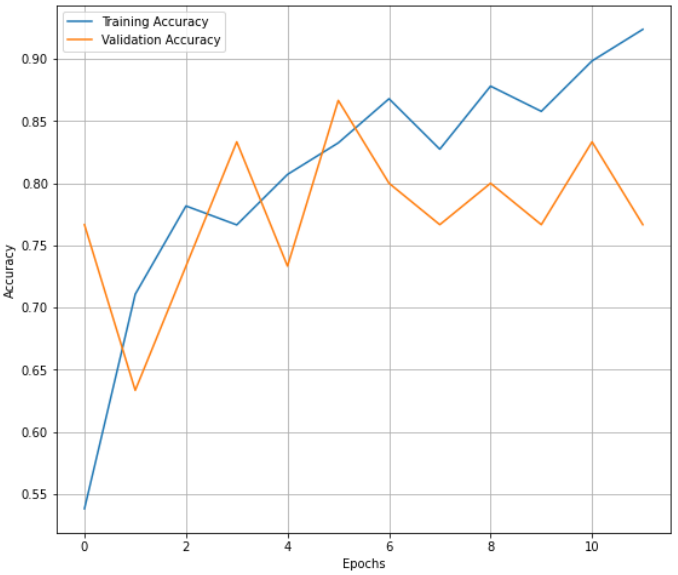
Chapter 4
Result and Discussion

The test of projected technique to discover and segment brain tumor is performed using MR images of diverse long-suffering. Each test image has brain tumor of diverse size, shape and intensity. Manual examination is used to check the correctness of automated segmented tumor area. The experimental result for different MR images containing tumor of different shapes, sizes and intensities.Our Dataset contains tumor and non-tumor MRI images and collected from different online resources. Radiopaedia contains real cases of patients, tumor images were obtained from Radiopaedia and Brain Tumor Image Segmentation Benchmark. In this work, efficient automatic brain tumor detection is performed by using convolution neural network. Simulation is performed by using python language. The accuracy is calculated and compared with the all other state of arts methods. The training accuracy, validation accuracy and validation loss are calculated to find the efficiency of proposed brain tumor classification scheme. In the existing technique, the Support Vector Machine (SVM) based classification is performed for brain tumor detection. It needs feature extraction output. Based on feature value, the classification output is generated and accuracy is calculated. The computation time is high and accuracy is low in SVM based tumor and non-tumor detection. In the proposed CNN based classification doesn’t require feature extraction steps separately. The feature value is taken from CNN itself. shows the classified result of Tumor and Non-tumor brain image. Hence the complexity and computation time is low and accuracy is high. The output of brain tumor classification accuracy is given in. Finally, the classification results as Tumor brain or non-tumor brain based on the probability score value. The normal brain image has the lowest probability score. Tumor brain has highest probability score value, when compared to normal and tumor brain.

The reason behind using various colors in segmentation is to identify an area of interest from the MRI images; human eyes are more sensitive to color images than the grey scale images. So here we are using the intensities of MRI images to place different color on the image the resultant image of this process will give the idea of tumor region.

In skull masking we have generated some horizontal, vertical, diagonal and anti-diagonal mask and after dividing image in small segments we have applied this masks over image sub parts and result image get combined is given this resultant image





Testing model performance:

```
[[ 9  3]
 [ 1 17]]
--------------------------
              precision    recall  f1-score   support

         0.0       0.90      0.75      0.82        12
         1.0       0.85      0.94      0.89        18

    accuracy                           0.87        30
   macro avg       0.88      0.85      0.86        30
weighted avg       0.87      0.87      0.86        30

--------------------------
0.8666666666666667
```

Chapter 5
Conclusion

The main goal of this research work is to design efficient automatic brain tumor classification with high accuracy, performance and low complexity In the conventional brain tumor classification is performed by using Fuzzy C Means (FCM) based segmentation, texture and shape feature extraction and SVM and DNN based classification are carried out. The complexity is low. But the computation time is high meanwhile accuracy is low. Further to improve the accuracy and to reduce the computation time, a convolution neural network based classification is introduced in the proposed scheme. Also the classification results are given as tumor or normal brain images. CNN is one of the deep learning methods, which contains sequence of feed forward layers. Also python language is used for implementation. Image net database is used for classification. It is one of the pre-trained models. So the training is performed for only final layer. Also raw pixel value with depth, width and height feature value are extracted from CNN. Finally, the Gradient decent based loss function is applied to achieve high accuracy. The training accuracy, validation accuracy and validation loss are calculated.

REFERENCES

1. Francis Galton, "Personal identification and description," In Nature,pp. 173-177, June 21, 1888.
2. W. Zaho, "Robust image based 3D face recognition," Ph.D. Thesis, Maryland University, 1999.
3. R. Chellappa, C.L. Wilson and C. Sirohey, "Humain and machine recognition of faces: A survey," Proc. IEEE, vol. 83, no. 5, pp. 705-740, may1995.
4. T. Fromherz, P. Stucki, M. Bichsel, "A survey of face recognition," MML Technical Report, No 97.01, Dept. of Computer Science, University of Zurich, Zurich, 1997.
5. T. Riklin-Raviv and A. Shashua, "The Quotient image: Class based recognition and synthesis under varying illumination conditions," In CVPR,P. II: pp. 566-571,1999.
6. G.j. Edwards, T.f. Cootes and C.J. Taylor, "Face recognition using active appearance models," In ECCV, 1998.
7. T. Sim, R. Sukthankar, M. Mullin and S. Baluja, "Memory-based face recognition for vistor identification," In AFGR, 2000.
8. T. Sim and T. Kanade, "Combing models and exemplars for face recognition: An illuminating example," In Proceeding Of Workshop on Models Versus Exemplars in Computer Vision, CUPR 2001.
9. L. Sirovitch and M. Kirby, "Low-dimensional procedure for the characterization of human faces," Journal of the Optical Society of America A, vol. 2, pp. 519-524, 1987.
10. M. Turk and A. Pentland "Face recognition using eigenfaces," In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.586- 591, 1991.
11. P. Belhumeur, P. Hespanha, and D. Kriegman, "Eigenfaces vs fisherfaces: Recognition using class specific linear projection," IEEE Transactions on Pattern Analysis and MachineIntelligence, vol. 19, no. 7, pp. 711-720, 1997.
12. M. Fleming and G. Cottrell, "Categorization of faces using unsupervised feature extraction," In Proc. IEEE IJCNN International Joint Conference on Neural Networks, pp. 65-70, 1990.
13. B. Moghaddam, W. Wahid, and A. Pentland, "Beyond eigenfaces:

14. Mazumdar; Sankar K. Pal, eds. (2012). Perception and Machine Intelligence: First Indo- Japan Conference, PerMIn 2012, Kolkata, India, January 12–13, 2011, Proceedings. Springer Science & Business Media. p. 29. ISBN 9783642273865.
15. Wechsler, Harry (2009). Malay K. Kundu; Sushmita Mitra (eds.). Reliable Face Recognition Methods: System Design, Implementation and Evaluation. Springer Science & Business Media. pp. 11–12. ISBN 9780387384641
16. Jun Wang; Laiwan Chan; DeLiang Wang, eds. (2012). Neural Information Processing: 13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006, Proceedings, Part II. Springer Science & Business Media. p. 198. ISBN 9783540464822.
17. Wechsler, Harry (2009). Reliable Face Recognition Methods: System Design, Implementation and Evaluation. Springer Science & Business Media. p. 12. ISBN 9780387384641.
18. Wechsler, Harry (2009). Malay K. Kundu; Sushmita Mitra (eds.). Reliable Face Recognition Methods: System Design, Implementation and Evaluation. Springer Science & Business Media. p. 12. ISBN 9780387384641.
19. Malay K. Kundu; Sushmita Mitra; Debasis Mazumdar; Sankar K. Pal, eds. (2012). Perception and Machine Intelligence: First Indo-Japan Conference, PerMIn 2012, Kolkata, India, January 12–13, 2011, Proceedings. Springer Science & Business Media. p. 29. ISBN 9783642273865.
20. "Mugspot Can Find A Face In The Crowd – Face-Recognition Software Prepares To Go To Work In The Streets". *ScienceDaily*. November 12, 1997. Retrieved November 6, 2007.
21. Malay K. Kundu; Sushmita Mitra; Debasis Mazumdar; Sankar K. Pal, eds. (2012). Perception and Machine Intelligence: First Indo-Japan Conference, PerMIn 2012, Kolkata, India, January 12–13, 2011, Proceedings. Springer Science & Business Media. p. 29. ISBN 9783642273865.
22. Li, Stan Z.; Jain, Anil K. (2005). *Handbook of Face Recognition*. Springer Science & Business Media. pp. 14–15. ISBN 9780387405957.
23. Kumar Datta, Asit; Datta, Madhura; Kumar Banerjee, Pradipta (2015). *Face Detection and Recognition: Theory and Practice*. CRC.

p. 123. ISBN 9781482226577.
24. Li, Stan Z.; Jain, Anil K. (2005). *Handbook of Face Recognition*. Springer Science & Business Media. p. 1. ISBN 9780387405957.
25. Li, Stan Z.; Jain, Anil K. (2005). *Handbook of Face Recognition*. Springer Science & Business Media. p. 2. ISBN 9780387405957.
26. "Airport Facial Recognition Passenger Flow Management". *hrsid.com*.
27. Bonsor, K. (September 4, 2001). "How Facial Recognition Systems Work". Retrieved June 2, 2008.
28. Smith, Kelly. "Face Recognition" (PDF). Retrieved June 4, 2008.
29. R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Trans. on PAMI, 1993, (15)10:1042–1052
30. R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009 ([1]TM bo
31. Zhang, David; Jain, Anil (2006). Advances in Biometrics: International Conference, ICB 2006, Hong Kong, China, January 5– 7, 2006, Proceedings. Berlin: Springer Science & Business Media. p. 183. ISBN 9783540311119.
32. "A Study on the Design and Implementation of Facial Recognition Application System". *International Journal of Bio-Science and Bio-Technology*.
33. Harry Wechsler (2009). Reliable Face Recognition Methods: System Design, Implementation and Evaluation. Springer Science & Business Media. p. 196. ISBN 9780387384641.
34. Jump up to:

a b c d Williams, Mark. "Better Face-Recognition Software". Retrieved June 2, 2008.

Crawford, Mark. "Facial recognition progress report". SPIE Newsroom. Retrieved October 6, 2011.