**Multiple Types of Cancer Classification Using CT/MRI Images Based on Learning Without Forgetting Powered Deep Learning Models**

**ABSTRACT:**

Cancer is the second biggest cause of death worldwide, accounting for one of every six deaths. On the other hand, early detection of the disease significantly improves the chances of survival. The use of Artificial Intelligence (AI) to automate cancer detection might allow us to evaluate more cases in less time. In this research, AI-based deep learning models are proposed to classify the images of eight kinds of cancer, such as lung, brain, breast, and cervical cancer. This work evaluates the deep learning models, namely Convolutional Neural Networks (CNN), against classifying images with cancer traits. Pre-trained CNN variants such as MobileNet, VGGNet, and DenseNet are employed to transfer the knowledge they learned with the ImageNet dataset to detect different kinds of cancer cells. We use Bayesian Optimization to find the suitable values for the hyperparameters. However, transfer learning could make it so that models can no longer classify the datasets they were initially trained. So, we use Learning without Forgetting (LwF), which trains the network using only new task data while keeping the network’s original abilities. The results of the experiments show that the proposed models based on transfer learning are more accurate than the current state-of-the-art techniques.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS**  **LIST OF ABBREVIATIONS**  **LIST OF TABLES** | i  v  vii  xi  xii |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. OBJECTIVE   1.3 EXISTING SYSTEM  1.3.1 EXISTINGSYSTEM DISADVANTAGES  1.3.2 LITERATURE SURVEY  1.4 PROPOSED SYSTEM  1.4.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.2.2 MODULES EXPLANATION  2.2.3 MODULE DIAGRAM  2.2.4 GIVEN INPUTAND EXPECTED OUTPUT  2.3 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS |  |
| 4. | **CHAPTER 4 :SYSTEM DESIGN**  **4.1 general**  4.1.1 activity diagram  4.1.2 USE CASE DIAGRAM  4.1.3 DATA FLOW DIAGRAM  4.1.4 SEQUENCE DIAGRAM  4.1.5 COLLABORATION DIAGRAM  4.1.6 CLASS DIAGRAM  4.1.7 SYSTEM ARCHITECTURE  4.1.8 OBJECT DIAGRAM  4.1.9 STATE DIAGRAM  4.1.10 COMPONENT DIAGRAM  4.1.11 E-R DIAGRAM |  |
| 5. | **CHAPTER 5 :SOFTWARE SPECIFICATION**  5.1 general |  |
| 6. | **CHAPTER 6 :IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION |  |

|  |  |  |
| --- | --- | --- |
| 7. | **CHAPTER 7 :SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 :SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 :**  **APPLICATIONS AND FUTURE ENHANCEMENT**  9.1 GENERAL  9.2 APPLICATIONS  9.3 FUTURE ENHANCEMENTS |  |
| **10** | **CHAPTER 10 :**  10.1CONCLUSION  10.2 REFERENCES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 4.2 | Activity Diagram |  |
| 4.3 | Use case Diagram |  |
| 4.4 | Data flow diagram |  |
| 4.5 | Sequence diagram |  |
| 4.6 | Collaboration diagram |  |
| 4.7 | Class diagram |  |
| 4.8 | Architecture Diagram |  |
| 4.9 | State Diagram |  |
| 4.1 | Component Diagram |  |

**LIST OF SYSMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | Represents a collection of similar entities grouped together. |
| 2. | Association | name  Class B  Class A  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor | Class A  Class A  Class B  Class B | It aggregates several classes into a single classes. |
| 4. | Aggregation | Interaction between the system and external environment |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation  (uses) | uses | Used for additional process communication. |
| 6. | Relation  (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the processes. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | Final state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |
| 13. | Use case |  | Interact ion between the system and external environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which are a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors, etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**CHAPTER-1**

**INTRODUCTION**

Cancer stands as a formidable global health challenge, claiming one in every six lives and ranking as the second leading cause of death worldwide. Despite its widespread prevalence, the prospects of survival significantly improve with early detection. Leveraging the capabilities of Artificial Intelligence (AI) to automate cancer detection holds the promise of evaluating a larger volume of cases in less time. In this research endeavour, our focus revolves around the application of AI-based deep learning models for the classification of images depicting eight distinct types of cancer, including lung, brain, breast, and cervical cancer. Specifically, we explore the efficacy of Convolutional Neural Networks (CNN), employing pre-trained variants such as MobileNet, VGGNet, and DenseNet. To optimize the models, we utilize Bayesian Optimization to identify optimal hyperparameter values. However, the implementation of transfer learning introduces the challenge of potential loss in the ability of models to classify datasets they were initially trained on. To address this, we employ Learning without Forgetting (LwF), a technique that preserves the network's original abilities while training it exclusively on new task data. The experimental results demonstrate that our proposed models, grounded in transfer learning, exhibit superior accuracy compared to current state-of-the-art techniques. Furthermore, we showcase the effectiveness of LwF in proficiently classifying both new datasets and previously encountered datasets, affirming its role in enhancing model performance and adaptability in cancer detection.

**1.2 OBJECTIVE:**

The primary objectives of this research endeavour are twofold: first, to investigate the application of Artificial Intelligence (AI) through deep learning models for the automated classification of various cancer types based on medical imaging, and second, to assess the performance and efficacy of Convolutional Neural Networks (CNN) variants, including MobileNet, VGGNet, and DenseNet, in this context. The specific focus is on eight distinct types of cancer, encompassing lung, brain, breast, and cervical cancer. The overarching goal is to contribute to the advancement of early cancer detection methodologies, acknowledging the critical impact such advancements can have on improving survival rates. To achieve these objectives, the research employs transfer learning, utilizing pre-trained CNN variants to leverage knowledge acquired from the ImageNet dataset for the purpose of classifying cancer cells in medical images. Furthermore, Bayesian Optimization is employed to fine-tune hyperparameters, enhancing the models' classification performance.

**1.3 EXISTING SYSTEM:**

Before the proliferation of deep learning concepts, traditional methods were extensively utilized for cancer detection in medical imaging. Image processing techniques such as edge detection, employing Sobel and Canny edge detectors, were employed to identify boundaries and contours indicative of potential tumors. Texture analysis, incorporating statistical measures and Gabor filters, characterized the texture patterns within images. Feature extraction and selection methods, including histogram-based features and shape-based descriptors, quantified pixel intensity distributions and delineated geometric properties of potential tumors. Classical machine learning algorithms like Support Vector Machines, Decision Trees, and Random Forests were commonly applied for binary classification tasks, leveraging extracted features for cancer identification. Radiomics, involving the extraction of quantitative features from images, and rule-based expert systems also played roles in cancer detection. Computer-Aided Diagnosis (CAD) systems, integrating image processing and classical machine learning, assisted radiologists in interpreting medical images. While these traditional approaches demonstrated efficacy in certain scenarios, the advent of deep learning, particularly Convolutional Neural Networks (CNNs), has ushered in a new era by allowing automatic learning of hierarchical representations directly from raw data, minimizing the need for manual feature engineering and significantly improving accuracy in cancer detection tasks.

**1.4 LITERATURE SURVEY**

**Title:** Multi-domain interventions to prevent cognitive impairment, Alzheimer’s disease, and dementia: From FINGER to world-wide FINGERS.

**Author:** A. Rosenberg, F. Mangialasche, T. Ngandu, A. Solomon

**Year:** 2020.

**Description:**

Alzheimer's disease (AD) and dementia are a global public health priority, and prevention has been highlighted as a pivotal component in managing the dementia epidemic. Modifiable risk factors of dementia and AD include lifestyle-related factors, vascular and metabolic disorders, and psychosocial factors. Randomized controlled clinical trials (RCTs) are needed to clarify whether modifying such factors can prevent or postpone cognitive impairment and dementia in older adults. Given the complex, multi factorial, and heterogeneous nature of late-onset AD and dementia, interventions targeting several risk factors and mechanisms simultaneously may be required for optimal preventive effects. The Finnish Geriatric Intervention Study to Prevent Cognitive Impairment and Disability (FINGER) is the first large, long-term RCT to demonstrate that a multi domain lifestyle-based intervention ameliorating vascular and lifestyle-related risk factors can preserve cognitive functioning and reduce the risk of cognitive decline among older adults at increased risk of dementia. To investigate the multi domain intervention in other populations and diverse cultural and geographical settings, the World-Wide FINGERS (WW-FINGERS) network was recently launched (https://alz.org/wwfingers). Within this network, new FINGER-type trials with shared core methodology, but local culture and context-specific adaptations, will be conducted in several countries. The WW-FINGERS initiative facilitates international collaborations, provides a platform for testing multi domain strategies to prevent cognitive impairment and dementia, and aims at generating high-quality scientific evidence to support public health and clinical decision-making. Furthermore, the WW-FINGERS network can support the implementation of preventive strategies and translation of research findings into practice.

**Title:** The estimated cost of dementia in japan, the most aged society in the world.

**Author:** M. Sado, A. Ninomiya, R. Shikimoto, B. Ikeda, T. Baba, K. Yoshimura, and M. Mimura.

**Year:** 2023.

**Description**: Objective Dementia has become a global critical issue. It is estimated that the global cost of dementia was 818 billion USD in 2015. The situation in Japan, which is the most aged country in the world, should be critical. However, the societal cost of dementia in Japan has not yet been estimated. This study was designed to estimate cost of dementia from societal perspective. Design we estimated the cost from societal perspective with prevalence based approach. Setting, participants and measures Main data sources for the parameters to estimate the costs are the National Data Base, a nationwide representative individual-level database for healthcare utilization, the Survey of Long-Term Care Benefit Expenditures, a nationwide survey based on individual-level secondary data for formal long-term care utilization, and the results of an informal care time survey for informal care cost. We conducted the analyses with ‘probabilistic modeling’ using the parameters obtained to estimate the costs of dementia. We also projected future costs. Results The societal costs of dementia in Japan in 2014 were estimated at JPY 14.5 trillion (se 66.0 billion). Of these, the costs for healthcare, long-term care, and informal care are JPY 1.91 trillion (se 4.91 billion), JPY 6.44 trillion (se 63.2 billion), and JPY 6.16 trillion (se 12.5 billion) respectively. The cost per person with dementia appeared to be JPY5.95 million (se 27 thousand). The total costs would reach JPY 24.3 trillion by 2060, which is 1.6 times higher than that in 2014. Conclusions the societal cost of dementia in Japan appeared to be considerable. Interventions to mitigate this impact should be considered.

**Title:** Deep Learning for Health Informatics

**Author:** D. Ravi, C. Wong, and F. Deligianni.

**Year:** 2021.

**Description:**

With a massive influx of multimodality data, the role of data analytics in health informatics has grown rapidly in the last decade. This has also prompted increasing interests in the generation of analytical, data driven models based on machine learning in health informatics. Deep learning, a technique with its foundation in artificial neural networks, is emerging in recent years as a powerful tool for machine learning, promising to reshape the future of artificial intelligence. Rapid improvements in computational power, fast data storage and parallelization have also contributed to the rapid uptake of the technology in addition to its predictive power and ability to generate automatically optimized high-level features and semantic interpretation from the input data. This article presents a comprehensive up-to-date review of research employing deep learning in health informatics, providing a critical analysis of the relative merit and potential pitfalls of the technique as well as its future outlook. The paper mainly focuses on key applications of deep learning in the fields of translational bioinformatics, medical imaging, pervasive sensing, medical informatics and public health.

**Title:**  Ensemble of 3D densely connected convolutional network for diagnosis of mild cognitive impairment and Alzheimer’s disease.

**Author:** H. Wang, Y. Shen, S. Wang, T. Xiao, L. Deng, and X. Wang.

**Year:** 2022.

**Description**:

Automatic diagnosis of Alzheimer’s disease (AD) and mild cognition impairment (MCI) from 3D brain magnetic resonance (MR) images plays an important role in early treatment of [dementia](https://www.sciencedirect.com/topics/neuroscience/dementia) disease. Deep learning architectures can extract potential features of [dementia](https://www.sciencedirect.com/topics/neuroscience/dementia) disease and capture brain anatomical changes from MRI scans. This paper proposes an ensemble of 3D densely connected [convolutional networks](https://www.sciencedirect.com/topics/computer-science/convolutional-network) (3D-DenseNets) for AD and MCI diagnosis. First, dense connections were introduced to maximize the information flow, where each layer connects with all subsequent layers directly. Then probability-based fusion method was employed to combine 3D-DenseNets with different architectures. Extensive experiments were conducted to analyze the performance of 3D-DenseNet with different hyper-parameters and architectures. Superior performance of the proposed model was demonstrated on ADNI dataset.

**Title:** A robust brain MRI classification with GLCM features.

**Author**: S. Jafarpour, Z. Sedghi, and M. C. Amirani.

**Year:** 2021**.**

**Description:**

Automated and accurate classification of brain MRI is such important that leads us to present a new robust classification technique for analyzing magnetic response images. The proposed method consists of three stages, namely, feature extraction, dimensionality reduction, and classification. We use gray level co-occurrence matrix (GLCM) to extract features from brain MRI and for selecting the best features, PCA+LDA is implemented. The classifiers goal is to classify subjects as normal and abnormal brain MRI. A classification with a success of 100% for two normal and abnormal classes is obtained by the both classifiers based on artificial neural network (ANN) and k-nearest neighbor (k-NN). The proposed method leads to a robust and effective technique, which reduces the computational complexity, and the operational time compared with other recent works.

**1.5 PROPOSED SYSTEM**

The proposed system in the context of the given abstract involves the application of Artificial Intelligence (AI) through deep learning models for the automated classification of various types of cancer based on medical imaging. Specifically, Convolutional Neural Networks (CNN) variants, including MobileNet, VGGNet, and DenseNet, are utilized for their ability to learn intricate features from images. The focus is on eight distinct types of cancer, such as lung, brain, breast, and cervical cancer. Transfer learning is employed, leveraging pre-trained CNN variants that have acquired knowledge from the ImageNet dataset to effectively detect different types of cancer cells. To optimize the models, Bayesian Optimization is applied to identify suitable hyperparameter values, enhancing the classification performance. The research also addresses a common challenge associated with transfer learning, which is the potential degradation of the model's ability to classify the datasets it was initially trained on. To mitigate this, the proposed system incorporates Learning without Forgetting (LwF). LwF ensures that the network maintains its original abilities while being trained exclusively on new task data, thus preserving its proficiency in classifying both new datasets and those encountered previously. The overall goal of the proposed system is to advance early cancer detection methodologies, contributing to improved survival rates by employing state-of-the-art AI techniques in the analysis of medical images.

**Key Modules:**

**Data Preprocessing Module:**

Responsible for cleaning and preparing the medical imaging dataset for training and testing.Involves tasks such as image normalization, resizing, and handling missing or noisy data.

**Image Classification Module**:

Utilizes Convolutional Neural Networks (CNNs) for image classification. Incorporates various CNN variants like MobileNet, VGGNet, and DenseNet to evaluate their performance in classifying different types of cancer.

**Transfer Learning Module:**

Implements transfer learning techniques to leverage pre-trained CNN models (MobileNet, VGGNet, DenseNet) that have acquired knowledge from the ImageNet dataset. Aims to adapt the pre-trained models for accurate cancer cell detection.

**Hyperparameter Optimization Module:**

Integrates Bayesian Optimization to fine-tune hyperparameters of the CNN models. Seeks to identify optimal parameter values that enhance the classification performance of the models.

**Learning without Forgetting (LwF) Module:**

Addresses the challenge of potential degradation in the model's ability to classify datasets it was initially trained on during transfer learning. Implements Learning without Forgetting to retain and enhance the original abilities of the neural network while training it exclusively on new task data.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL:**

The proposed project aims to develop an advanced system for cancer detection leveraging Artificial Intelligence (AI) and deep learning models. Cancer remains a significant global health challenge, and early detection is crucial for improving survival rates. The project focuses on the classification of various types of cancer using medical imaging data, employing Convolutional Neural Networks (CNNs) such as MobileNet, VGGNet, and DenseNet. Transfer learning is incorporated to adapt pre-trained models from the ImageNet dataset to the task of detecting different cancer cells. Hyperparameter optimization is employed to fine-tune the models, and Learning without Forgetting (LwF) is utilized to address potential degradation in the models' ability to classify datasets they were initially trained on. The project includes modules for data preprocessing, image classification, transfer learning, hyperparameter optimization, LwF implementation, performance evaluation, results analysis, and optional user interface development. Comprehensive documentation and ethical considerations are integral parts of the project, ensuring transparency, accountability, and responsible use of the developed models. The overall goal is to contribute to the advancement of early cancer detection methodologies, potentially surpassing current state-of-the-art techniques and providing a robust framework for future research in this critical healthcare domain.

**2.2 METHODOLOGIES**

**2.2.1MODULES NAME:**

**MODULE:**

* **Dataset**
* **Importing the necessary libraries**
* **Retrieving the images**
* **Splitting the dataset**
* **Building the model**
* **Apply the model and plot the graphs for accuracy and loss**
* **Accuracy on test set**
* **Saving the Trained Model**

**MODULES DESCSRIPTION:**

**1 Dataset:**

In the first module, we developed the system to get the input dataset for the training and testing purpose. We have taken the dataset from https://www.kaggle.com/datasets/obulisainaren/multi-cancer.

1. **Importing the necessary libraries:**

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy , matplotlib and tensorflow.

**3 Retrieving the images:**

We will retrieve the images and their labels. Then resize the images to (176X208) as all images should have same size for recognition. Then convert the images into numpy array.

**4 Splitting the dataset:**

Split the dataset into train and test. 80% train data and 20% test data.

**5 Building the model:**

Nowadays, there are already several CNN models that have been released publicly. These Models has a very deep layer and trained using computers that have high specifications (most of which stand out are their GPU and RAM). One of those models that we will discuss here is VGG19.

**A.VGG19**

VGG is an innovative object-recognition model that supports up to 19 layers. Built as a deep CNN, VGG also outperforms baselines on many tasks and datasets outside of Image Net. VGG is now still one of the most used image-recognition architectures.

**6. Apply the model and plot the graphs for accuracy and loss:**

We will compile the model and apply it using fit function. The batch size will be 2. Then we will plot the graphs for accuracy and loss. We got an average training accuracy of 97.2%.

**7. Accuracy on test set:**

We got an accuracy of 94.02% on test set

**8. Saving the Trained Model:**

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into .h5 file

Make sure you have pickle installed in your environment.

Next, let’s import the module and dump the model into.h5 file

**2.3 TECHNIQUE USED OR ALGORITHM USED**

* + 1. **EXISTING TECHNIQUE:**
* Deep learning is a subfield of machine learning and a collection of algorithms that are inspired by the structure of human brain and try to imitate the functions of human brain, which is the reason these algorithms are most of the times also termed as ‘‘neural networks”.
* These algorithms are called ‘‘deep” as the input passes through series of non-linear transformations before it becomes output.
* Convolution neural network (CNN) is one such deep learning algorithm in which the transformations are done using an operation called ‘‘convolution.
* VGG16 was trained on very general images of Image Net dataset (natural images), it was still able to extract useful features for our classification task.
* We also computed various metrics to support the performance of our classification model and compared accuracy of our method with past methods.

**2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED**

* Vgg19, VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers.
* The “deep” refers to the number of layers with VGG-19 consisting of 16 and 19 convolutional layers. The VGG architecture is the basis of ground breaking object recognition models.
* VGG19 is an advanced CNN with pre-trained layers and a great understanding of what defines an image in terms of shape, color, and structure.
* VGG19 is very deep and has been trained on millions of diverse images with complex classification tasks.
* VGG-19 is an extension of the original VGGNet, and the "19" in its name refers to the total number of layers, including convolutional and fully connected layers.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

The interpretation of the handwriting character by developing techniques and methods such as improvement of character classification techniques. The accurate and rapid classification for accurate information retrieval, sound classification, stock price forecasting.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system do and not how it should be implemented.

# Processor - Pentium –IV

* Speed - 1.1 GHz
* Ram - 256 MB
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

**MINIMUMSYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* PROCESSOR : Pentium i3 Processor
* RAM : 2GB DD RAM
* HARD DISK : 250 GB

**SOFTWARE REQUIREMENTS**

* BACK END : PYTHON
* OPERATING SYSTEM : WINDOWS 7
* IDE : Spyder3

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, This paper has explained about a library named as Sparx which provides the solution for a programmer to acquire cleaned data for further analysis. Sparx is an exclusive data-preprocessing library, which involves transforming raw data into a machine-understandable format. The intention behind developing Sparx was to build a better, automated data-preprocessing library.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

**The major non-functional Requirements of the system are as follows**

**Usability**

The system is designed with completely automated process hence there is no or less user intervention.

**Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using python is more reliable.

**Performance**

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

**Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

**Implementation**

The system is implemented in web environment using Django framework. The server is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Django provides web application.

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

**4.2 UML DIAGRAMS**

**4.2.1 USE CASE DIAGRAM**

Login

CNN(VGG19)

Classification

Image acquisition

Image segmentation

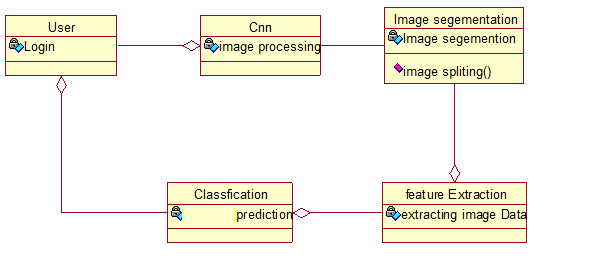
User

Feature extraction

**EXPLANATION:**

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. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

**4.2.2 CLASS DIAGRAM**

****

**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

**4.2.3 OBJECT DIAGRAM**



**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.2.4 COMPONENT DIAGRAM**



**EXPLANATION**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.2.5 DEPLOYMENT DIAGRAM**

CNN(VGG19)

User Login

Image

Acquisition

Image Segmentation

Feature Extraction

Classification

**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**4.2.6 SEQUENCE DIAGRAM**

UserLogin

CNN(VGG19)

Image

acquistion

Image

segementation

Feature

extraction

Classfication

Login

image acquisition

image split

extract data

Cancer Type Prediction

return prediction to user

**EXPLANATION:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.2.7 COLLABORATION DIAGRAM**

Classfication

User

Login

CNN

(VGG19)

Image

acquisition

Image

segmentation

Feature

Extraction

1: Login

2:

3: image acquisition

5: extract data

6: Cancer Type Prediction

7: return prediction to user

**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.2.8 STATE DIAGRAM**

User Login

CNN(VGG19)

Image acquisition

Image segmentation

Feature extraction

Conclusion

**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. **4.2.9 ACTIVITY DIAGRAM**

User Login

CNN(VGG19 )

Image acquisition

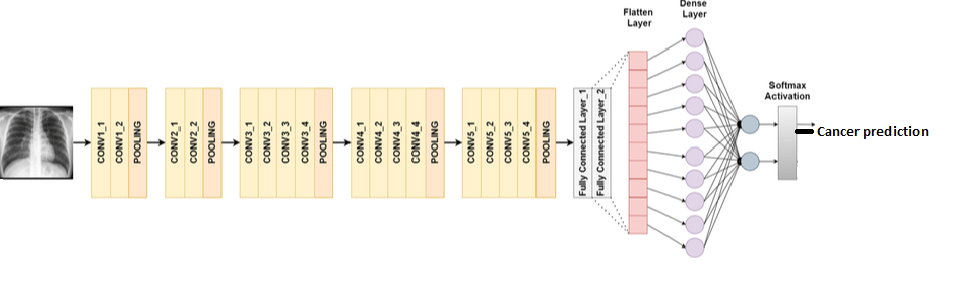
Image Segmentation

Classification

**EXPLANATION:**

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

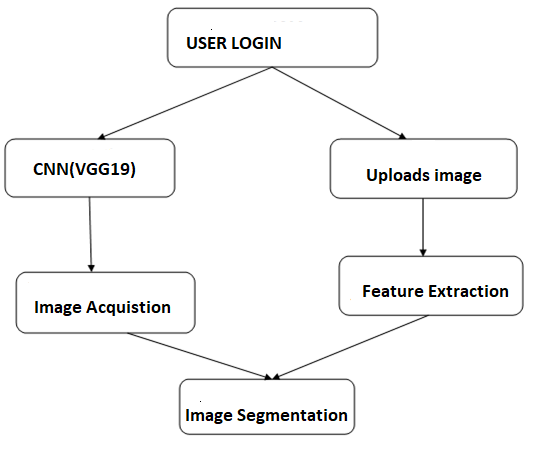
**4.3 SYSTEM ARCHITECTURE**

****

**VGG19 Architecture for Classification of multiple types of cancer prediction**

**Flow Chat Diagram**

**Level-0**

****

**Level-01**

**Classification**

**Cervical Cancer**

**Brain Cancer**

**Breast Cancer**

**Lung Cancer**

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

## History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

#### Importance of Python

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### Features of Python

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are

Listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**Libraries used in python:**

* numpy - mainly useful for its N-dimensional array objects.
* pandas - Python data analysis library, including structures such as data frames.
* matplotlib - 2D plotting library producing publication quality figures.
* scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.



Figure: NumPy, Pandas, Matplotlib, Scikit-learn

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

**CODING:**

**CHAPTER 7**

**SNAPSHOTS**

**7.1 SNAPSHOTS**

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

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.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3TYPES OF TESTS**

**8.3.1 UNIT TESTIN**

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

**8.3.2 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.

**8.3.3 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.

**8.3.4 PERFORMANCE TEST**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 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.

**8.3.6 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.

**ACCEPTANCE TESTING FOR DATA SYNCHRONIZATION:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 BUILD THE TEST PLAN**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**APPLICATIONS AND FUTURE ENHANCEMENT**

**9.1 General:**

The utilization of pre-trained variants like MobileNet, VGGNet, and DenseNet, coupled with Bayesian Optimization for hyperparameter tuning, has demonstrated remarkable efficacy in enhancing the accuracy of cancer detection. The incorporation of Learning without Forgetting (LwF) as a mitigation strategy for potential loss in model performance during transfer learning has proven to be instrumental in preserving the network's original capabilities while adapting to new tasks. Our experimental results highlight the superiority of our proposed models over current state-of-the-art techniques, affirming their potential for real-world applications in early cancer detection**.**

**9.2 Applications:**

The successful implementation of AI-based deep learning models for cancer detection, as outlined in this project, holds great potential for various real-time applications in the field of healthcare. Some of the general real-time applications include:

**Early Cancer Detection:** Deploying the developed models in real-time can aid in the early detection of various types of cancer, such as lung, brain, breast, and cervical cancer. Early diagnosis is critical for improving treatment outcomes and increasing the chances of survival.

**Diagnostic Assistance:** The AI models can serve as valuable tools for healthcare professionals by providing assistance in the diagnosis of cancer. By rapidly analyzing medical images, the system can help identify potential cancerous lesions or abnormalities, allowing for quicker decision-making in a clinical setting.

**Radiology Support:** Integrating the AI models into radiology workflows can enhance the efficiency of radiologists. The system can assist in interpreting medical imaging data, allowing radiologists to focus on more complex cases and improving overall diagnostic accuracy.

**Screening Programs:** The developed models can be employed in population-wide screening programs for certain types of cancer. This can facilitate the identification of individuals at risk, enabling early intervention and preventive measures.

**Telemedicine and Remote Consultations:** In the context of telemedicine, the AI models can be utilized for remote consultations. Medical practitioners can share medical images with the system in real-time, receiving automated insights that contribute to timely decision-making, especially in areas with limited access to specialized healthcare services.

**Clinical Decision Support Systems:** Integration into clinical decision support systems can provide healthcare professionals with real-time recommendations based on the analysis of medical images. This can contribute to more informed and efficient decision-making during patient care.

**Training and Education:** The developed models can be used as educational tools for training healthcare professionals, allowing them to familiarize themselves with diverse cancer cases and enhancing their diagnostic skills.

**9.3 FUTURE ENHANCEMENT:**

For future enhancements, further exploration and refinement of deep learning architectures, possibly incorporating more advanced models or ensemble techniques, could contribute to even higher accuracy rates. Additionally, the integration of multimodal data, such as combining imaging data with clinical and genetic information, may offer a more comprehensive understanding of cancer pathology. Continuous updates to the training dataset to include diverse and evolving cases will be crucial for maintaining model relevance and generalizability. Furthermore, exploring real-time implementation of the developed models in clinical settings, considering factors like interpretability and explainability, will be pivotal for the seamless integration of AI in routine medical practices. Ultimately, ongoing collaboration with medical professionals and researchers will be essential for refining and validating the proposed models for widespread adoption, marking a significant step forward in the global battle against cancer.

**CHAPTER 10**

**CONCLUSIONAND REFERENCES**

**10.1 CONCLUSION**

In conclusion, our research underscores the significant potential of leveraging Artificial Intelligence, specifically deep learning models such as Convolutional Neural Networks (CNNs), for the automated classification of diverse cancer types. The utilization of pre-trained variants like MobileNet, VGGNet, and DenseNet, coupled with Bayesian Optimization for hyperparameter tuning, has demonstrated remarkable efficacy in enhancing the accuracy of cancer detection. The incorporation of Learning without Forgetting (LwF) as a mitigation strategy for potential loss in model performance during transfer learning has proven to be instrumental in preserving the network's original capabilities while adapting to new tasks. Our experimental results highlight the superiority of our proposed models over current state-of-the-art techniques, affirming their potential for real-world applications in early cancer detection**.**

**10.2 REFERENCES**

[1] Top Opportunities for Artificial Intelligence to Improve Cancer Care. Accessed: Nov. 29, 2021. [Online]. Available: https://healthitanalytics.com/features/top-opportunities-for-artificial-intelligence-to-improve-cancer-care

[2] M. Subramanian, K. Shanmugavadivel, and P. S. Nandhini, ‘‘On finetuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves,’’ Neural Comput. Appl., vol. 34, no. 16, pp. 13951–13968, Aug. 2022.

[3] M. Subramanian, ‘‘Hyperparameter optimization for transfer learning of VGG16 for disease identification in corn leaves using Bayesian optimization,’’ Big Data, vol. 10, no. 3, pp. 215–229, Jun. 2022.10352 VOLUME 11, 2023 M. Subramanian et al.: Multiple Types of Cancer Classification Using CT/MRI Images

[4] S. Krishnamoorthy, A. Shanthini, G. Manogaran, V. Saravanan, A. Manickam, and R. D. J. Samuel, ‘‘Regression model-based feature filtering for improving hemorrhage detection accuracy in diabetic retinopathy treatment,’’ Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.,

vol. 29, no. 1, pp. 51–71, Apr. 2021.

[5] S. Roy, T. Meena, and S.-J. Lim, ‘‘Demystifying supervised learning in healthcare 4.0: A new reality of transforming diagnostic medicine,’’ Diagnostics, vol. 12, no. 10, p. 2549, Oct. 2022.

[6] S. Krishnamoorthy, Y. Zhang, S. Kadry, and W. Yu, ‘‘Framework to segment and evaluate multiple sclerosis lesion in MRI slices using VGGUNet,’’ Comput. Intell. Neurosci., vol. 2022, pp. 1–10, Jun. 2022.

[7] S. Rezayi, N. Mohammadzadeh, H. Bouraghi, S. Saeedi, and A. Mohammadpour, ‘‘Timely diagnosis of acute lymphoblastic leukemia using artificial intelligence-oriented deep learning methods,’’ Comput. Intell. Neurosci., vol. 2021, pp. 1–12, Nov. 2021.

[8] S. R. Gunasekara, H. N. T. K. Kaldera, and M. B. Dissanayake, ‘‘A systematic approach for MRI brain tumor localization and segmentation using deep learning and active contouring,’’ J. Healthcare Eng., vol. 2021, pp. 1–13, Feb. 2021.

[9] V. K. Reshma, N. Arya, S. S. Ahmad, I. Wattar, S. Mekala, S. Joshi, and D. Krah, ‘‘Detection of breast cancer using histopathological image classification dataset with deep learning techniques,’’ BioMed Res. Int., vol. 2022, pp. 1–13, Mar. 2022.

[10] S. Zhao, Y. He, J. Qin, and Z. Wang, ‘‘A semi-supervised deep learning method for cervical cell classification,’’ Anal. Cellular Pathol., vol. 2022, pp. 1–12, Feb. 2022.

[11] M. Pedersen, M. B. Andersen, H. Christiansen, and N. H. Azawi, ‘‘Classification of renal tumour using convolutional neural networks to detect on cocytoma,’’ Eur. J. Radiol., vol. 133, Dec. 2020, Art. no. 1093