**BRAIN MRI IMAGE ALZHEIMER CLASSIFIER**

**ABSTRACT**

Many different architectures of Convolutional Neural Networks (CNNs) have been developed for use in classifying images and recognizing objects. When it comes to image-based categorization, handling hundreds of MRI image slices that are essentially identical across patients is a challenging task for CNN. By utilizing a 2D CNN design, it becomes difficult to confidently categorize a large number of individuals as having Alzheimer's disease, mild cognitive impairment, or normal cognition. To solve this problem, we have streamlined the concept of patient classification based on 3D MRI while still giving due credit to the 2D features derived from the CNN framework. Here, we share our approach to extracting 2D features from MRI scans in a format that can be used in a classification system. Our experiment demonstrates the outcome of categorizing 3 patient participants into 2 groups. After reducing the dimensionality of a 2D image with principal component analysis and truncated sparse encoding (PCA+TSNE), we used a convolutional neural network (CNN) to extract generic features for classification. Despite the lackluster performance, this seems to be an improvement over probability-based categorization using a CNN that was trained from scratch. The created feature is highly malleable and can be fine-tuned to improve precision, responsiveness, and specificity.

**INTRODUCTION**

There has been a lot of work done to process, simulate, and interpret the results of medical examinations using various imaging techniques such as MRI (Magnetic Imaging Resonance), PET (Positronemission tomography), and Computed Tomography (CT) scans. This is being done for the purpose of Computer Aided Diagnosis (CAD), which will be of vital importance to medical professionals. In a similar vein, multiple researchers have conducted a variety of studies using MRI as the primary biomarker in order to effectively construct a Computer Assisted Diagnosis (CAD) system for the diagnosis and detection of Alzheimer's disease.

Alzheimer's disease (AD), Parkinson's disease (PD), Huntington's disease (HD), and other neurodegenerative disorders are all included under the umbrella term neuro-degenerative diseases, which mostly affect the older population. For example, Alzheimer's disease is one of the most frequent types of dementia. Several of the symptoms of Alzheimer's disease are well-known, including problems sleeping, memory loss, and dramatic shifts in mood. According to research that was published by the World Health Organization (WHO), there are roughly 40 million people around the world who are afflicted with AD.

It is anticipated that this number will reach 135 million by the year 2050. In spite of the fact that Alzheimer's disease cannot be cured, early diagnosis and appropriate treatment of the condition can slow down the rate at which neurons are lost. This is an essential aspect of the disease that should not be overlooked. According to the data presented in the WHO report, there is an urgent requirement for the research and development of computer-aided diagnostics systems for the prompt diagnosis of AD. This encourages the researchers to direct their attention more intently towards the creation of cutting-edge diagnostic tools and algorithms for the early detection and classification of Alzheimer's disease.

Based on statistical movements of the MR image intensities, demonstrated a method for distinguishing healthy controls from people with Alzheimer's disease (AD). For the purpose of picture classification, fuzzy logic is utilized as an algorithmic classification method. In the beginning, the regions of interest were given to the fuzzy interference system as an input function. These regions were determined based on the statistical motions of the MR images, which had 24 different intensities. When calculating the area under the curve, the k-fold cross validation method is the one that is utilized.

The system's performance efficiency can be represented by this AUC calculation that was made. In addition to this, the maximum number of feature shifts that are necessary to achieve optimal AUC was studied and mentioned. Outlined the methods and advantages of the Gaussians Mixture Model, which is utilized for the classification of images to diagnose Alzheimer's disease (AD). Images of the diffusion tensor are given a classification using the GMM in order to locate the AD. In the work that is being proposed, GMM is used in conjunction with functional anisotropy (FA) and mean diffusivity (MD) in order to conduct an analysis.

Comparisons are made between the GMM and the linear discriminant performance analyses. According to the findings, the generalized estimating equation (GMM) provides a higher level of accuracy than the linear discriminant technique.

**LITERATURE SURVEY**

1. **Samsuddin Ahmed, Kyu Yeong Choi, et al “Ensembles of Patch-Based Classifiers for Diagnosis of Alzheimer Diseases”2019-IEEE Access**

Traditional machine learning methods are being used to automatically diagnose Alzheimer's disease (AD), while deep learning-based methods are becoming prominent. State-of-the-art multimodal diagnosis methods outperform manual diagnosis. Unfortunately, gathering data from diverse modalities is time-consuming, expensive, and may have radioactive adverse effects. Structural magnetic resonance imaging is our focus (sMRI). Our goal: 1) to boost accuracy to match state-of-the-art approaches, 2) to overcome the overfitting problem, and 3) to assess brain landmarks that yield AD diagnosis features. We targeted the left and right hippocampuses here. First, we use ensembles of simple convolutional neural networks (CNNs) as feature extractors and softmax cross-entropy as the classifier. Given data paucity, we used a patch-based strategy. Gwangju's National Research Center for Dementia (GARD) cohort dataset was used for our project. We manually located the left and right hippocampus and supplied the CNN three view patches (TVPs) after preprocessing. 90.05% accuracy. Our model performed comparably to state-of-the-art approaches on the same dataset.

1. **Xingyu Gao, Feng Shi, et al “Task-Induced Pyramid and Attention GAN for Multimodal Brain Image Imputation and Classification in Alzheimer's Disease”2022- IEEE Journal of Biomedical and Health Informatics**

Medical imaging technologies like MRI and PET can detect subtle brain anatomical and functional changes, making brain disorders like Alzheimer's disease easier to diagnose (AD). As PET is expensive or unavailable, multimodal pictures may be incomplete. Most approaches removed missing data, reducing the sample size. Multimodal feature extraction and combination remain difficult. We present a deep learning system for multimodal brain image imputation and classification using a task-induced pyramid and attention generative adversarial network (TPA-GAN) and a pathwise transfer dense convolution network (PT-DCN). First, we present a TPA-GAN with pyramid convolution, attention module, and illness classification task to generate missing PET data from MRI. Using imputed multimodal pictures, we design a dense convolution network with pathwise transfer blocks to gradually learn and incorporate multimodal features for disease categorization. Our technique outperforms state-of-the-art image imputation and brain illness diagnosis using ADNI-1/2 datasets.

1. **Mona Ashtari-Majlan, Abbas Seifi, et al “A Multi-Stream Convolutional Neural Network for Classification of Progressive MCI in Alzheimer’s Disease Using Structural MRI Images”2022- IEEE Journal of Biomedical and Health Informatics**

Although some individuals with progressing MCI will become Alzheimer's disease, early detection of Alzheimer's disease and its prodromal stage, also known as mild cognitive impairment (MCI), is crucial. For the purpose of differentiating between static MCI and progressive MCI, we present a multi-stream deep convolutional neural network fed with patch-based imaging data. To begin, we use a multivariate statistical test to compare MRI scans of people with Alzheimer's disease and cognitively normal patients in order to find different anatomical landmarks. To identify MRI scans, the proposed multi-stream convolutional neural network is supplied with extracted patches based on these landmarks. To make up for the dearth of progressive MCI training data, we next train the architecture in a distinct scenario utilising samples from Alzheimer's disease images, which are physically comparable to the ones of progressive MCI, and cognitively normal images. Lastly, we fine-tune the model using progressive MCI and stable MCI data by transferring the trained model weights to the suggested architecture. Based on experimental results using the ADNI-1 dataset, we may conclude that our method achieves a higher F1-score (85.96%) than competing methods for classifying patients with MCI.

1. **Emtiaz Hussain, Mahmudul Hasan, et al “Deep Learning Based Binary Classification for Alzheimer’s Disease Detection using Brain MRI Images”2020-15th IEEE Conference on Industrial Electronics and Applications (ICIEA)**

Alzheimer’s disease is an irremediable, ongoing brain ailment that gradually damages memory and thinking skills and, finally, the ability to carry out the simplest tasks. Globally, it's a major disease. Alzheimer's is incurable. Machine learning techniques, especially deep learning-based Convolutional Neural Network (CNN), are utilised to improve the procedure for the detection of Alzheimer’s disease. CNN has excelled in MRI image analysis and biomedical research recently. CNN-based brain MRI study has been done to detect Alzheimer's disease. The suggested CNN model was not properly compared to pre-trained CNN models (InceptionV3, Xception, MobilenetV2, VGG). Hence, we provide a 12-layer CNN model for binary classification and Alzheimer's disease diagnosis using brain MRI data. Using the Open Access Series of Imaging Studies (OASIS) dataset, the proposed model is compared to existing CNN models in accuracy, precision, recall, F1 score, and ROC curve. The main contribution of the research is a 12-layer CNN model with an accuracy of 97.75%, which is greater than any other existing CNN models reported on this dataset. The publication also compares our model to pre-trained CNN models (InceptioV3, Xception, MobilenetV2, VGG). The proposed model outperforms existing models in experiments.

1. **Tripti Goel, Rahul Sharma, et al “Multimodal Neuroimaging based Alzheimer's Disease Diagnosis using Evolutionary RVFL Classifier”2023- IEEE Journal of Biomedical and Health Informatics**

Consistent decline in mental faculties over time is a hallmark of Alzheimer's disease (AD), one of the most well-known forms of dementia. Mild cognitive impairment is a progressive disease that can only be reversed via early diagnosis and treatment (MCI). MRI and PET scans can detect the most frequent indicators used to diagnose Alzheimer's disease, including structural atrophy and the formation of plaques and tangles. Since early identification of this fatal neurodegenerative illness relies on incorporating structural and metabolic information, this paper suggests wavelet transform-based multimodality fusion of MRI and PET data. Additionally, the deep learning model, ResNet-50, retrieves the fused images' features. The collected features are then classified using a single-hidden-layer random-vector functional link (RVFL). An evolutionary method is being used to fine-tune the weights and biases of the original RVFL network. The effectiveness of the proposed algorithm is demonstrated by extensive testing and comparison using the ADNI dataset, which is available to the public.

1. **Emimal Jabason, M. Omair Ahmad, et al “Classification of Alzheimer’s Disease from MRI Data Using an Ensemble of Hybrid Deep Convolutional Neural Networks”2019-IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)**

Although there is no cure for Alzheimer's disease (AD), an accurate early diagnosis is extremely important for both the patient and social care, and it will become even more significant once disease-modifying agents are available to prevent, cure, or even slow down the progression of the disease. In recent years, classification of AD through deep learning techniques has been one of the most active research areas in the medical field. However, most of the existing techniques cannot leverage the entire spatial information; hence, they lose the inter-slice correlation. In this paper, we propose a novel classification algorithm to discriminate patients having AD, mild cognitive impairment (MCI), and cognitively normal (CN) using an ensemble of hybrid deep learning architectures to leverage a more complete spatial information from the MRI data. The experimental results obtained by applying the proposed algorithm on the OASIS dataset show that the performance of the proposed classification framework to be superior to that of some conventional methods.

1. **Amir Ebrahimi-Ghahnavieh, Suhuai Luo, et al “Transfer Learning for Alzheimer's Disease Detection on MRI Images” 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)**

In this article, we use deep learning methods to MRI images in an effort to detect Alzheimer's disease. A significant difficulty in this area of study is the dearth of appropriate data for training a deep model. Based on our research, we know that subject classification using 2D convolutional neural networks and transfer learning is one of the current trends in Alzheimer's disease research. Each 3D MRI volume is then separated into 2D image slices such that a previously trained 2D convolutional neural network may be used to perform segmentation and classification on each slice individually. Nevertheless, the 2D convolutional neural network cannot take into account the interconnectedness of the 2D picture slices that make up an MRI volume. We suggest using a recurrent neural network after a convolutional neural network to learn the association between image sequences for each subject and to reach a conclusion using all input slices rather than just one. Our findings demonstrate that improving the accuracy of the entire system by training the recurrent neural network on features extracted by a convolutional neural network is possible.

1. **Huan Lao, Xuejun Zhang, et al “Regression and Classification of Alzheimer’s Disease Diagnosis Using NMF-TDNet Features From 3D Brain MR Image”2022- IEEE Journal of Biomedical and Health Informatics**

With deep learning and medical imaging technology, many researchers use convolutional neural network (CNN) to obtain deep-level medical image features to better classify Alzheimer's disease (AD) and predict clinical scores. The lightweight deep-learning network PCANet uses principal component analysis (PCA) to generate multilevel filter banks for centralised sample learning, binarizes, and generates blockwise histograms to obtain image features. PCANet's flexibility is reduced by the tens of thousands or hundreds of thousands of extracted PCANet features and the sample data-dependent formation of multilevel filter banks. This paper proposes the nonnegative matrix factorization tensor decomposition network, a data-independent network based on PCANet, to solve these issues (NMF-TDNet). Instead of PCA, we utilise nonnegative matrix factorization (NMF) to generate multilevel filter banks for sample learning, then use the learning results to build a higher-order tensor and perform tensor decomposition (TD) to reduce data dimensionality and provide picture features. Finally, our method uses these features as SVM input for AD classification diagnosis and clinical score prediction. Our method is extensively tested on ADNI-1, ADNI-2, and OASIS datasets. NMF-TDNet features as input outperformed PCANet features in data dimensionality reduction.

1. **Chiyu Feng, Ahmed Elazab, et al “Deep Learning Framework for Alzheimer’s Disease Diagnosis via 3D-CNN and FSBi-LSTM”2019- IEEE Access**

Neurodegenerative Alzheimer's disease (AD) is irreversible. Mild cognitive impairment (MCI) is the prodromal stage of AD, which can be progressive (pMCI) or stable (i.e., sMCI). Deep learning has helped convolutional neural networks (CNNs) recognise MRI and PET images for AD diagnosis. CNNs are difficult to employ for AD diagnosis due to the lack of imaging data. We create a new deep learning framework for this. Our approach takes advantage of 3D-CNN and FSSBi-LSTM. Initially, we create a 3D-CNN architecture for MRI and PET deep feature representation. FSBi-LSTM is then applied to deep feature maps' hidden spatial information to increase performance. The ADNI dataset validates our method. Our method beats comparable algorithms in differentiating AD, pMCI, and sMCI from NC with average accuracies of 94.82%, 86.36%, and 65.35%, respectively.

1. **Ahmad Waleed Salehi, Preety Baglat, et al “A CNN Model: Earlier Diagnosis and Classification of Alzheimer Disease using MRI” 2020-International Conference on Smart Electronics and Communication (ICOSEC)**

The most common form of dementia, Alzheimer's Disease (AD), damages brain cells and impairs memory and daily function. Utilizing MRI (Magnetic Resonance Imaging) scan brain images, we may use AI technology to diagnose, predict, and classify AD patients as having or not having this lethal disease. This is done to create the finest prediction and detection tools for radiologists, doctors, and carers to save time, money, and treat patients with this disease. DL algorithms operate well with huge datasets, making them beneficial for AD diagnosis in recent years. We used 1512 mild, 2633 normal, and 2480 AD MRI images from the ADNI 3 class to implement Convolutional Neural Network (CNN) for early diagnosis and classification of AD. The model performed well compared to another relevant research with 99% accuracy. We also compared the result with our previous work on OASIS dataset using machine learning algorithms, which showed that deep learning approaches may be better for large amounts of medical data.

**EXISTING SYSTEM**

Some of the symptoms of dementia include memory loss, trouble processing language, difficulties communicating, and changes in mood. The principal impacts of the condition on a patient's mind and body go hand in hand with a significant adjustment in lifestyle and the ability to carry out ordinary duties. The voxel, region, and patch-based techniques relied significantly on manually constructed features and feature representations since they saw segmentation jobs as classification problems. It took more time and a larger number of expertly segmented images to train classification algorithms.

**DISADVANTAGES**

* minimal foresight.
* Existing method takes longer time to classify.

**PROPOSED SYSTEM**

Preprocessing of MR images, skull stripping (the extraction of essential parts of the brain by removing the brain membrane and unnecessary tissues), feature extraction and selection of non-redundant features from the processed MR images, application of fractal analysis on the selected features, and classification using the proposed machine learning method are all required for successful detection and classification of Alzheimer's disease from brain MR images.

**SYSTEM REQUIREMENTS**

**Hardware Requirements**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

* System: Intel core I3 processer 64 bits.
* Monitor: LED.
* Mouse: Logitech.
* Ram: 4.00 GB.

**Software Requirements**

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

* Operating system: Windows 64 bit
* Language: Python
* Platform: Anaconda3

**DATAFLOW DIAGRAM**

**START**

**DATASET COLLECTION**

**PREPROCESSING**

**TRAINING &VALIDATION**

**Very\_Mild\_Demented**

**Non\_Demented**

**Moderate\_Demented**

**Mild\_Demented**

**MODEL PREDICTION**

**FEATURE EXTRACTION**

**PREDICTED OUTPUT**

**BLOCK DIAGRAM**

**PREDICTED OUTPUT**

**FEATURE EXTRACTION**

**MODEL PREDICTION**

**CLASSIFIER**

**TRAINING &VALIDATION**

**PREPROCESSING**

**DATASET COLLECTION**

**START**

**SEQUENCE DIAGRAM**

**Start**

**Model Pridiction**

**Training**

**Preprocessing**

**Dataset collection Feature Extraction**

**Mild**

**Moderate**

**Predicted output**

**Very\_Mild**

**Non\_demented**

**USECASE DIAGRAM**

**LANGUAGE SPECIFICATION**

**PYTHON:**

Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

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.

**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.

**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.

**Python Features**

Python's features include −

**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.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

**Development**

Python's development is conducted largely through the Python Enhancement Proposal (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues and documenting Python design decisions. Python coding style is covered in PEP 8. Outstanding PEPs are reviewed and commented on by the Python community and the steering council.

Enhancement of the language corresponds with development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are discussed in the Roundup bug tracker maintained at python.org. Development originally took place on a self-hosted source-code repository running Mercurial, until Python moved to GitHub in January 2017.

CPython's public releases come in three types, distinguished by which part of the version number is incremented:

* Backward-incompatible versions, where code is expected to break and need to be manually ported. The first part of the version number is incremented. These releases happen infrequently for example, version 3.0 was released 8 years after 2.0.
* Major or "feature" releases, about every 18 months, are largely compatible but introduce new features. The second part of the version number is incremented. Each major version is supported by bugfixes for several years after its release.
* Bugfix releases, which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release.

Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented.

Many alpha, beta, and release-candidates are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large unit test suite during development, and using the BuildBot continuous integration system.

The community of Python developers has also contributed over 86,000 software modules (as of 20 August 2016) to the Python Package Index (PyPI), the official repository of third-party Python libraries.

The major academic conference on Python is PyCon. There are also special Python mentoring programmes, such as Pyladies

**SOFTWARE SPECIFICATION**

**ANACONDA**



Anaconda is an amazing collection of scientific Python packages, tools, resources, and IDEs. This package includes many important tools that a Data Scientist can use to harness the incredible force of Python. Anaconda individual edition is free and open source. This makes working with Anaconda accessible and easy. Anaconda has grown an exceptionally large community. Anaconda makes it easy to connect to several different scientific, Machine Learning and Data Science packages.

**The key features:**

* Neural Networks
* Machine Learning
* Predictive Analytics
* Data Visualization
* Bias Mitigation

**What is Anaconda?**

Anaconda is a free open-source data science tool that focusses on the distribution of R and Python programming languages for data science and machine learning tasks. Anaconda aims at simplifying the data management and deployment of the same.

Anaconda is a powerful data science platform for data scientists. The package manager of Anaconda is the conda which manages the package versions.

Anaconda is a tool that offers all the required package involved in data science at once. The programmers choose Anaconda for its ease of use.

Anaconda is written in Python, and the worthy information on Conda is unlike pip in Python, this package manager checks for the requirement of the dependencies and installs it if it is required. More importantly, warning signs are given if the dependencies already exist.

Conda very quickly installs the dependencies along with frequent updates. It facilitates creation and loading with equal speed along with easy environment switching.

The installation of Anaconda is very easy and most preferred by non-programmers who are data scientists.

Anaconda is pre-built with more than 1500 Python or R data science packages. Anaconda has specific tools to collect data using Machine learning and Artificial Intelligence.

Anaconda is indeed a tool used for developing, testing and training in one single system. The tool can be managed with any project as the environment is easily manageable.

Anaconda is great for deep models and neural networks. You can build models, deploy them, and integrate with leading technologies in the subject. Anaconda is optimized to run efficiently for machine learning tasks and will save you time when developing great algorithms. Over 250 packages are included in the distribution. You can install other third-party packages through the Anaconda terminal with conda install. With over 7500 data science and machine learning packages available in their cloud-based repository, almost any package you need will be easily accessible. Anaconda offers individual, team, and enterprise editions. Included also is support for the R programming language.

The Anaconda distribution comes with packages that can be used on Windows, Linux, and MacOS. The individual edition includes popular package names like numpy, pandas, scipy, sklearn, tensorflow, pytorch, matplotlib, and more. The Anaconda Prompt and PowerShell make working within the filesystem easy and manageable. Also, the GUI interface on Anaconda Navigator makes working with everything exceptionally smooth. Anaconda is an excellent choice if you are looking for a thriving community of Data Scientists and ever-growing support in the industry. Conducting Data Science projects is an increasingly simpler task with the help of great tools like this.

**Creating virtual environment**

Like many other languages Python requires a different version for different kind of applications. The application needs to run on a specific version of the language because it requires a certain dependency that is present in older versions but changes in newer versions. Virtual environments make it easy to ideally separate different applications and avoid problems with different dependencies. Using virtual environment we can switch between both applications easily and get them run. There are multiple ways of creating an environment using virtualenv, venv and conda. Conda command is preferred interface for managing installations and virtual environments with the Anaconda Python distribution.

**Anaconda Navigator**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

* JupyterLab
* Jupyter Notebook
* QtConsole
* Spyder
* Glue
* Orange
* RStudio
* Visual Studio Code

**FEASIBILITY STUDY:**

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

**Economic Feasibility**

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

**Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

**Operational Feasibility**

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

**SYSTEM TESTING**

**INTRODUCTION**

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

**TYPES OF TESTING**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

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

**Functional test**

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

Functional testing is centred on the following items:

* Valid Input: identified classes of valid input must be accepted.
* Invalid Input: identified classes of invalid input must be rejected.
* Functions: identified functions must be exercised.
* Output: identified classes of application outputs must be exercised.

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

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

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

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

**Unit Testing**

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

**Test strategy and approach**

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

**Test objectives**

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

**Features to be tested**

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

**Integration Testing**

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

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

**Test Results:**

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

**Acceptance Testing**

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

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

**CONCLUSION**

As a result, we put our theory of transfer learning from CNN to other classifiers through its paces, following the steps in sequential order. It is possible for us to draw one of the defining qualities for training the classifier is the ability to transfer learnt parameters and features from a CNN that has been trained. If the feature transformation, selection, and classification processes are carried out in an intelligent manner, the CNN features trained classifier can achieve a higher level of performance than CNN networks itself. The performance of CNN can even be enhanced by properly modifying and tuning the architecture of CNN, as well as by appropriately optimizing the classification system.

**REFERENCE**

1. Y. Zhang, S. Wang, and Z. Dong, “Classification of Alzheimer disease based on structural magnetic resonance imaging by kernel support vector machine decision tree”, Progress In Electromagnetics Research, vol. 144, January 2014.
2. S. Chaplot, L. M. Patnaik, and N. R. Jagannathan, “Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network”, Biomedical Signal Processing and Control, vol. 1, pp. 86-92, January 2006.
3. M. Maitra, and A. Chatterjee, “A Slantlet transform based intelligent system for magnetic resonance brain image classification”, Biomedical Signal Processing and Control, vol. 1, pp. 299-306, October 2006.
4. E. S. A. El-Dahshan, T. Hosny, and A. B. M. Salem, “Hybrid intelligent techniques for MRI brain images classification”, Digital Signal Processing, vol. 1, pp. 299-306, October 2006.
5. Y. Zhang, L. Wu, and S. Wang, “Magnetic resonance brain image classification by an improved artificial bee colony algorithm”, Progress In Electromagnetics Research, vol. 116, pp. 65-79, 2011.
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, Advances in neural information processing systems, pp. 1097- 1105, 2012.
7. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition” arXiv:1512.03385, 2015.
8. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”, In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1-9, 2015.
9. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014.
10. N. Tajbakhsh, J. Shin, S. Gurudu, C. Kendall, M. Gotway, and J. Liang, “Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?”, IEEE Transactions on Medical Imaging, vol. 35, pp. 1299-1312, May 2016.
11. H. Shin, H. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, and J. Yao, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics, and Transfer Learning”, IEEE Transactions on Medical Imaging, vol. 35, pp. 1285-1298, May 2016.
12. H. R. Roth, L. Lu, J. Liu, J. Yao, A.Seff, K. Cherry, L. Kim, and Ronald M. Summers, “Improving Computer-Aided Detection Using Convolutional Neural Networks and Random View Aggregation”, IEEE Transactions on Medical Imaging, vol. 35, pp. 1170-1181, May 2016.
13. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition”, Proc.IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pp. 512-519, 2014.