DETECTION OF BRAIN TUMORS USING HYBRID MODEL CONVOLUTIONAL NEURAL NETWORK AND

NEURAL AUTOREGRESSIVE DISTRIBUTION ESTIMATION

*Minor project report submitted*

*in partial fulfillment of the requirement for award of the degree of*

Bachelor of Technology in

Computer Science & Engineering

By

Y. VAIBHAV (18UECS0953)

*Under the guidance of*

*Mr. SARAN RAJ*

*Assistant Professor*



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr.SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956) Accredited by NAAC with A Grade

CHENNAI 600 062, TAMILNADU, INDIA, June,2021

CERTIFICATE

It is certified that the work contained in the project report titled ”A STUDY ON

DETECTION OF BRAINTUMOUR USING HYBRID NERUALNETWORK” by

”Y. VAIBHAV (18UECS0953)” has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor

Mr.S.Saran Raj

Assistant Professor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr.Sagunthala R&D

Institute of Science and Technology

June,2021

Signature of Head of the Department

Dr. V. Srinivasa Rao, M.Tech., Ph.D.,

Professor & Head

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr.Sagunthala R&D

Institute of Science and Technology

June,2021

DECLARATION

We declare that this written submission represents my ideas in our own words and where others’ ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Y.VAIBHAV

Date: / /

APPROVAL SHEET

This project report entitled SILHOUETTE ANALYSIS ON K- MEANS CLUSTERING by Y.VAIBHAV (18UECS0953) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners Supervisor

Mr.S.SARAN RAJ,M.E,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO).

DSc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.

Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved Vice Chancellor Prof. S. SALIVAHANAN, for providing us with an environment to complete our project successfully.

We record indebtedness to our Dean & Head, Department of Computer Science and Engineering Dr. V. SRINIVASA RAO, M.Tech., Ph.D., for immense care and encouragement towards us throughout the course of this project.

We also take this opportunity to express a deep sense of gratitude to Our Internal Supervisor Mr.s.SARAN RAJ,M.E,Assistant Professor, for her cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E., Ms.S.FLORENCE, M.Tech., for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

Y.VAIBHAV (18UECS0953)

# ABSTRACT

A Brain tumor is an abnormal growth of cell inside the skull malignant brain tumors are among the most . Dead full type of cancer with direct consequences such as cognitive decline and poor quality of life We are presenting detection of brain tumors using hybrid model convolution neural network and Neural Autoregressive Distribution Estimation (NADE) models, which are neural network architectures applied to the problem of unsupervised distribution and density estimation and hybrid convolution method They leverage the probability product rule and a weight sharing scheme inspired from restricted Boltzmann machines, to yield an estimator that is both tractable and has good generalization performance distribution of a emulator by the fixable distribution estimator to solve the problem various type of inference problem classification of a brain tumor region.

Keywords:Machine Learning, Deep Learning, Neural Networking, NADE, CNN

LIST OF FIGURES

Fig no Name of the Figure Page No

|  |  |
| --- | --- |
| 4.1 Architecture Diagram | 11 |
| 4.2 Data Flow Diagram | 12 |
| 4.3 Collabration Diagram | 13 |
| 4.4 Sequence Diagram | 14 |
| 6.4 Output 1 | 21 |
| 6.5 Output 3 | 21 |

# LIST OF ACRONYMS AND ABBREVIATIONS

ACRONYM EXPANSION

AI Artificial Intelligence

ML Machine Learning

Dp Deep learning

CNN Convolutional Neural Network

TABLE OF CONTENTS

Page.No

[ABSTRACT v](#_Toc25077)

[LIST OF ACRONYMS AND ABBREVIATIONS vii](#_Toc25078)

[1 INTRODUCTION 1](#_Toc25079)

[1.1 Introduction 1](#_Toc25080)

[1.2 Aim of the project 1](#_Toc25081)

[1.3 Project Domain 2](#_Toc25082)

[1.4 Scope of the Project 2](#_Toc25083)

[1.5 Methodology 2](#_Toc25084)

[2 LITERATURE REVIEW 4](#_Toc25085)

[3 PROJECT DESCRIPTION 6](#_Toc25086)

[3.1 Existing System 6](#_Toc25087)

[3.2 Proposed System 7](#_Toc25088)

[3.3 Feasibility Study 8](#_Toc25089)

[3.3.1 Economic Feasibility 8](#_Toc25090)

[3.3.2 Technical Feasibility 9](#_Toc25091)

[3.3.3 Social Feasibility 9](#_Toc25092)

[3.4 System Specification 9](#_Toc25093)

[3.4.1 Hardware Specification 9](#_Toc25094)

[3.4.2 Software Specification 10](#_Toc25095)

[3.4.3 Standards and Policies 10](#_Toc25096)

[4 MODULE DESCRIPTION 11](#_Toc25097)

[4.1 General Architecture 11](#_Toc25098)

[4.2 Design Phase 12](#_Toc25099)

[4.2.1 Data Flow Diagram 12](#_Toc25100)

[4.2.2 colabration Diagram 13](#_Toc25101)

[4.2.3 Sequence Diagram 14](#_Toc25102)

[5 IMPLEMENTATION AND TESTING 15](#_Toc25103)

[5.1 Input and Output 15](#_Toc25104)

[5.1.1 Input Design 15](#_Toc25105)

[5.1.2 Output Design 16](#_Toc25106)

[5.2 Testing 16](#_Toc25107)

[5.3 Types of Testing 16](#_Toc25108)

[5.3.1 Unit testing 16](#_Toc25109)

[5.3.2 Integration testing 17](#_Toc25110)

[5.3.3 Functional testing 17](#_Toc25111)

[5.3.4 White Box Testing 17](#_Toc25112)

[5.3.5 Black Box Testing 18](#_Toc25113)

[5.4 Testing Strategy 18](#_Toc25114)

[6 RESULTS AND DISCUSSIONS 19](#_Toc25115)

[6.1 Efficiency of the Proposed System 19](#_Toc25116)

[6.2 Comparison of Existing and Proposed System 19](#_Toc25117)

[6.3 Advantages of the Proposed System 19](#_Toc25118)

[6.4 Sample Code 21](#_Toc25119)

[7 CONCLUSION AND FUTURE ENHANCEMENTS 23](#_Toc25120)

[7.1 Conclusion 23](#_Toc25121)

[7.2 Future Enhancements 23](#_Toc25122)

[8 PLAGIARISM REPORT 24](#_Toc25123)

[9 SOURCE CODE & POSTER PRESENTATION 25](#_Toc25124)

[9.1 Source code 25](#_Toc25125)

[9.2 Poster Presentation 29](#_Toc25126)

[References 29](#_Toc25127)

Chapter 1

# INTRODUCTION

## 1.1 Introduction

Brain images analysis is a challenging and complicated task to detect brain tumors such as malignant brain tumors. In conflict with images due to noise and organ movements are other point in brain tumor detection. A deep neural network can directly get images,extracting features,and classifies the target output. CNN networks are deep learning structures with enormous number of parameters Networks face vanishing gradient problem due to a large number of layers,reducing the number of layers is a good approach to resolve but it will reduce the accuracy. To maintain high accuracy additional knowledge about datasets such as the density of data is needed that can be provided by neural autoregressive distribution estimation. By implementing NADE we can reduce significant number of convolutional layers compared to other CNNs. This method doesn’t have a preprocessing stage such as image segmentation or common filters for generating additional features.

## 1.2 Aim of the project

To detect brain tumors from MRI images base on deep learning using hybrid model CNN and NADE.

## 1.3 Project Domain

In our project we are Neural Autoregressive Distribution Estimation (NADE) models, which are neural network architectures applied to the problem of unsupervised distribution and density estimation.We also use Convolutional Neural Network (CNN or ConvNet) which consists of convolution, activation, max pooling, and fully connected layers which hasthe capability of extracting features and image classification.

## 1.4 Scope of the Project

To impact on next generation to detection more accuracy in image of brain tumoro detect brain tumors from MRI images base on deep learning using hybridmodel CNN and NADE.

## 1.5 Methodology

MODULE-1.4.1

Explanation about NADE

A Convolution NADE is an autoregressive model which uses convolutional layers instead of fully connected hidden layers. They leverage the probability product rule and a weight sharing scheme inspired from restricted Boltzmann machines, to yield an estimator that is both tractable and has good generalization performance.Although, CNN is capable of exploring features from raw high dimensional data and exploit patterns that lead the model to correct output. Studies show that it is possible to build a CNN with the autoregressive.Neural network specifically tailored to this task of density or distribution modeling.

MODULE-1.4.2

Explanation about CNN

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyze visual imagery. The ”neocognitron”was introduced by Kunihiko Fukushima in 1980. It was inspired by the work of Hubel and Wiesel. The neocognitron introduced the two basic types of layers in CNNs: convolutional layers, and downsampling layers Proposed an algorithm using backpropagation to learn Neocognitron which nowadays is known as CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume through a differentiable function. This network consists of convolution, activation, max pooling, and fully connected layers which has the capability of extracting features and image classification. Applications image recognition,video analysis,drug discovery etc

MODULE-1.4.3

Explanation about Final architecture validation

This brain tumor dataset containing 3064 T1-weighted contrast-inhanced images from 233 patients with three kinds of brain tumor: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices).In this dataset 70 of images are used in testing phase and 30 of images are used to get result phase. This model achives high performance in front of a few imbalanced classification brain tumor datasets with 94 accuracy after trained by a 6-fold cross-validation technique and adom optimizer.

Chapter 2

# LITERATURE REVIEW

[1]R. Tamilselvi

The brain is composed of nerve cells and supportive tissues such as glial cells and meninges.The projected MRI database is a termed BRAMSIT, characterized by an attempt to offer a group of normal and malignant brain tumor images.

[2]Raheleh Hashemzehia

A deep neural network can directly get images, extracting features, and classifies the target output.Density of data is needed that can be provided by neural autoregressive distribution estimators.

[3]Rav‘ı D, Wong C

Brain Tumor segmentation is one of the most crucial and arduous tasks in the terrain of medical image processing as a human-assisted manual classification can result in inaccurate prediction and diagnosis. Moreover, it is an aggravating task when there is a large amount of data present to be assisted. Brain tumors have high diversity in appearance and there is a similarity between tumor and normal tissues and thus the extraction of tumor regions from images becomes unyielding.

1. litjens G

Brain tumor classification plays an important role in clinical diagnosis and effective treatment. In this work, we propose a method for brain tumor classification using an ensemble of deep features and machine learning classifiers. In our proposed framework, we adopt the concept of transfer learning and uses several pre-trained deep convolutional neural networks.

1. S. Somasundaram

The blue circle represents a node or neuron from which the name ‘neural network’ is derived from. There is an input to each neuron. The arrows or ‘axons’ represent the connection between neurons. The result is an output which generates an approximation of the image which is iteratively refined.

Chapter 3

# PROJECT DESCRIPTION

## 3.1 Existing System

A Convolution NADE is an autoregressive model which uses convolutional layers instead of fully connected hidden layers. They leverage the probability product rule and a weight sharing scheme inspired from restricted Boltzmann machines, to yield an estimator that is both tractable and has good generalization performance. Although, CNN is capable of exploring features from raw high dimensional data and exploit patterns that lead the model to correct output. Studies show that it is possible to build a CNN with the autoregressive. Neural network specifically tailored to this task of density or distribution modeling. In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyze visual imagery. The ”recognition”was introduced by Kunihiko Fukushima in 1980. It was inspired by the work of Hubel and Wiesel. The recognition introduced the two basic types of layers in CNNs: convolutional layers, and down sampling layers Proposed an algorithm using back propagation to learn Recognition which nowadays is known as CNN A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume through a differentiable function. This network consists of convolution, activation, max pooling, and fully connected layers which has the capability of extracting features and image classification. Applications image recognition, video analysis, drug discovery etc This brain tumor dataset containing 3064 T1-weighted contrast-inhanced images from 233 patients with three kinds of brain tumor: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices).

In this dataset, 70 of images are used in testing phase and 30 of images are used to get result phase. This model achives high performance in front of a few imbalanced classification brain tumor datasets with 94 accuracy after trained by a 6-fold cross-validation technique and adom optimizer.

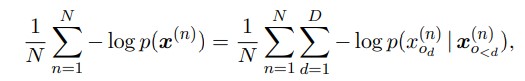
## 3.2 Proposed System

NADE

The NADE framework was first introduced for binary variables by Larochelle and Murray (2011), and concurrent work by Gregor and LeCun (2011). The framework was then generalized to real-valued observations (Uria et al., 2013), and to versions based on deep neural networks that can model the observations in any order (Uria et al., 2014). This paper pulls together an extended treatement of these papers, with more experimental results, including some by Uria (2015). We also report new work on modeling 2D images by incorporating convolutional neural networks into the NADE framework. For each type of data, we’re able to reach competitive results, compared to popular directed and undirected graphical model alternatives.

NADE begins with the observation that any D-dimensional distribution p(x) can be factored into a product of one-dimensional distributions, in any order o (a permutation of the integers 1, . . . , D).

NADE can be trained by maximum likelihood, or equivalently by minimizing the average negative log-likelihood .



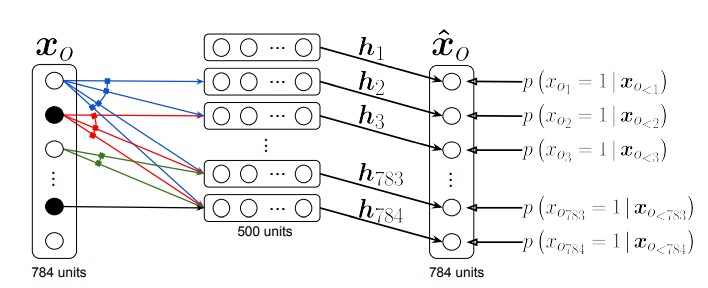


Illustration of a NADE model

In the input layer, units with value 0 are shown in black while units with value 1 are shown in white. The dashed border represents a layer pre-activation.The outputs xˆO give predictive probabilities for each dimension of a vector xO, given elements earlier in some ordering. There is no path of connections between an output and the value being predicted, or elements of xO later in the ordering. Arrows connected together correspond to connections with shared (tied) parameters.

## 3.3 Feasibility Study

### 3.3.1 Economic Feasibility

The system developed and installed will be good benefit to the organization. The system will be developed and operated in the existing hardware and software infrastructure. So there is no need of additional hardware and software for the system.

### 3.3.2 Technical Feasibility

The language used in the development is Python Anaconda Environment.Platform.The observer pattern along with existing pattern will be update the results eventually.System environment used in development is windows.

### 3.3.3 Social Feasibility

We present Neural Autoregressive Distribution Estimation (NADE) models, which areneural network architectures applied to the problem of unsupervised distribution and density estimation.By this project the user will understand how the data is distributed by using the NADE and CNN models.

## 3.4 System Specification

### 3.4.1 Hardware Specification

* Working framework: Windows 7 or more, mac OS, Linux.
* System design: 64-bit or 32-bit.
* RAM: 4GB or above.

### 3.4.2 Software Specification

* Anaconda
* Python
* jupyter notebook
* spyder compiler

### 3.4.3 Standards and Policies

The Standard and Policies of our project are:

IEEE 829 - Software Test Documentation

IEEE 830 -Software Requirements Specifications

IEEE1012- Software verification and validation

IEEE 1016 - Software design description

Chapter 4

# MODULE DESCRIPTION

## 4.1 General Architecture

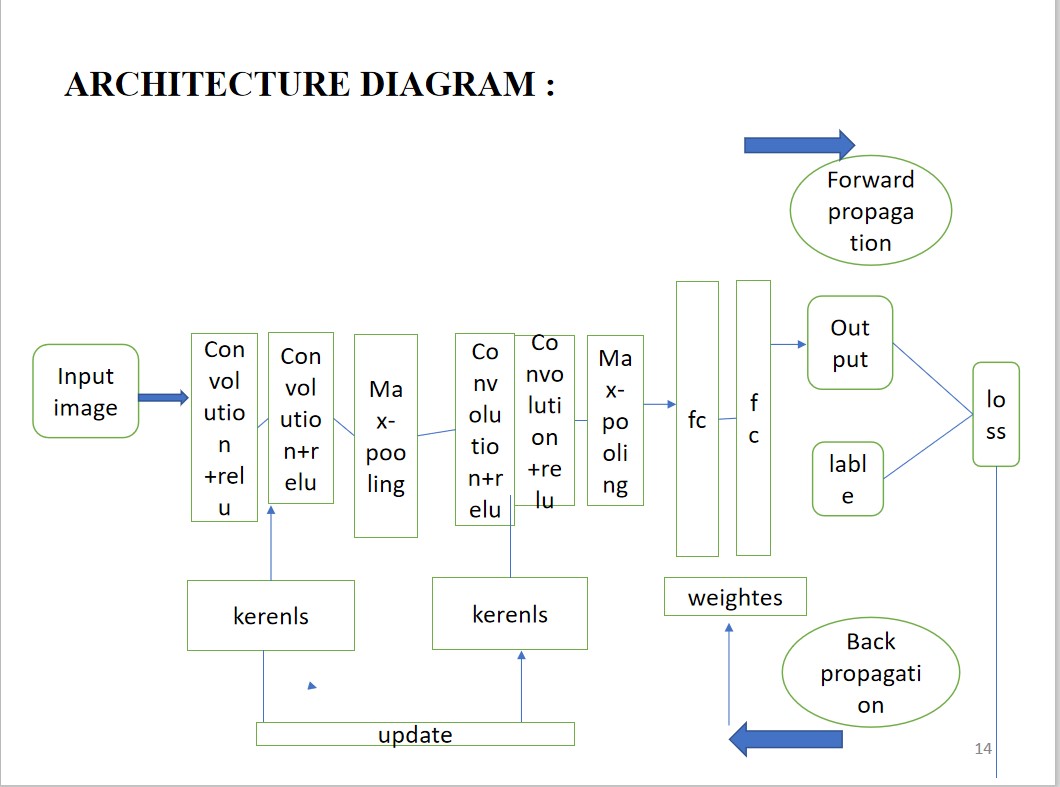


Fig 4.1:Architecture Diagram

## 4.2 Design Phase

### 4.2.1 Data Flow Diagram

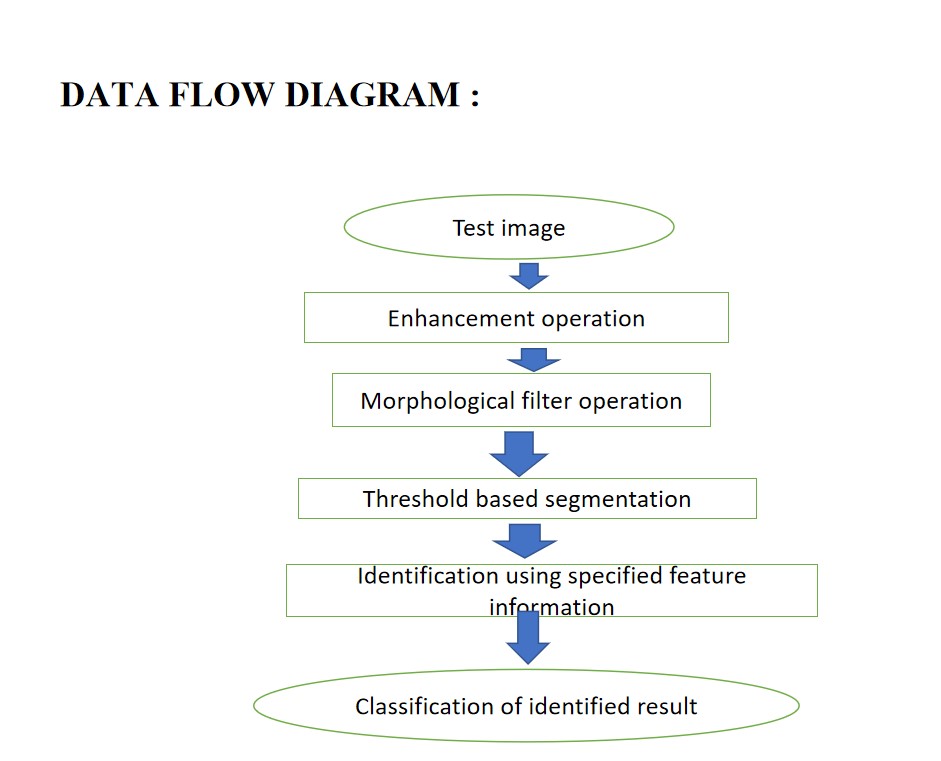


Fig 4.2:Data Flow Diagram

### 4.2.2 colabration Diagram

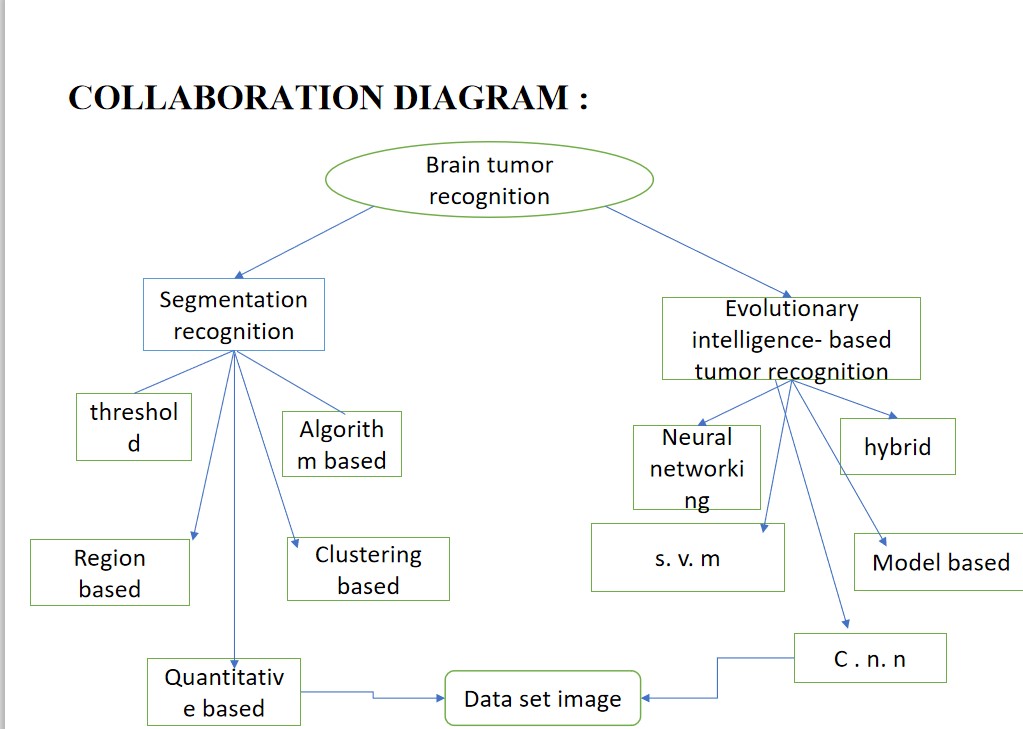


Fig 4.3: Colabration Diagram

### 4.2.3 Sequence Diagram

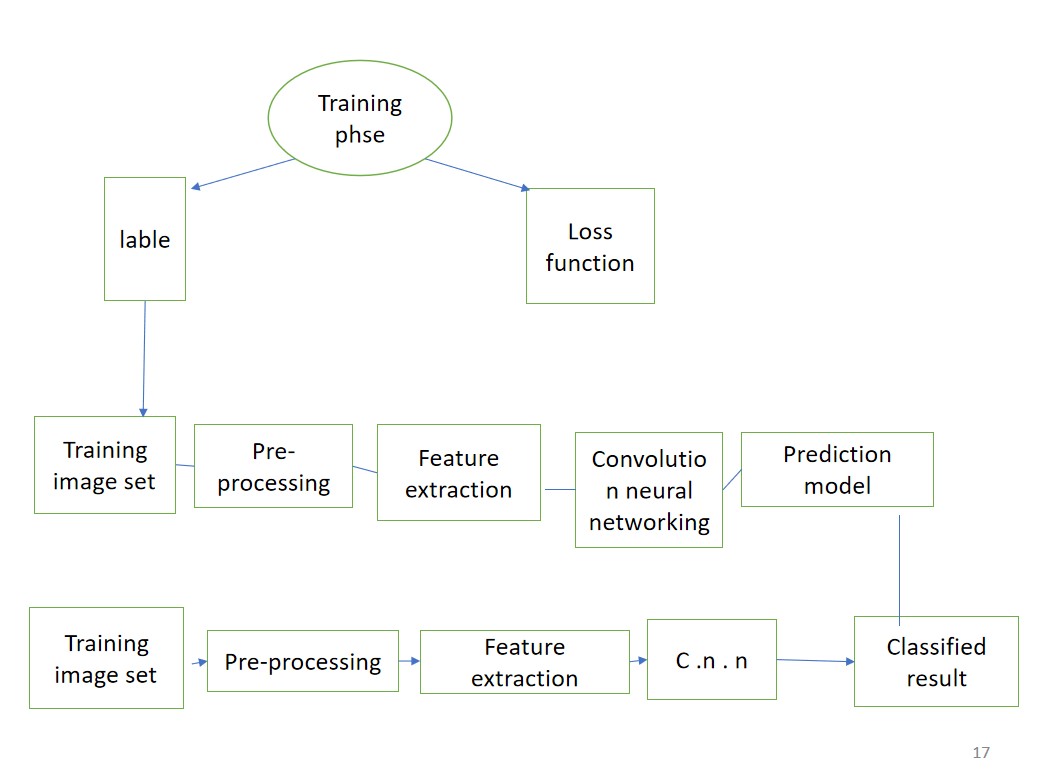


Fig 4.4:Sequence Diagram

Chapter 5

# IMPLEMENTATION AND TESTING

## 5.1 Input and Output

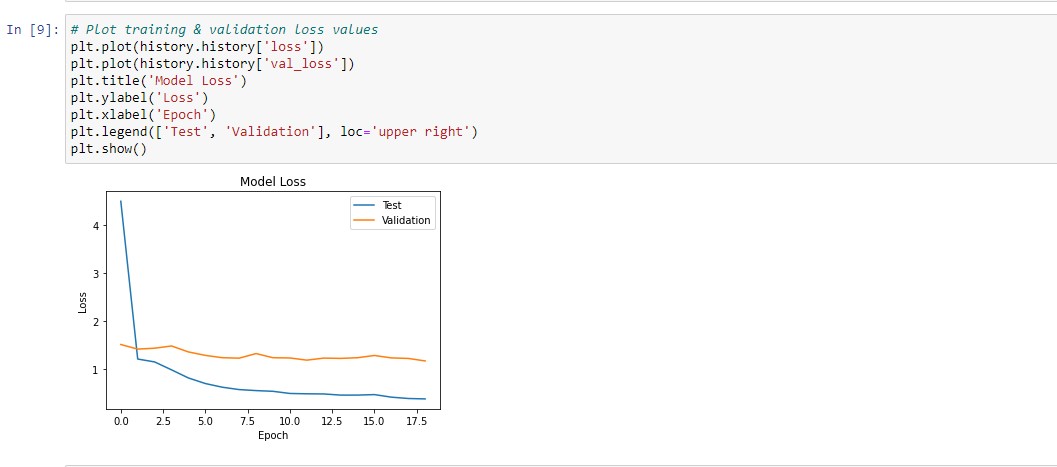
### 5.1.1 Input Design

Basic structure of CNN.Input Design: The input layer is the input of the whole CNN. In the neural network of image processing, it generally represents the pixel matrix of the image. Convolutional layer: The convolutional layer is used to extract image features. Low-level convolutional layer extracts shallow features (such as edges, lines, and corners).



### 5.1.2 Output Design

The output from the final (and any) Pooling and Convolutional Layer in model loss graph in test case and validation case.



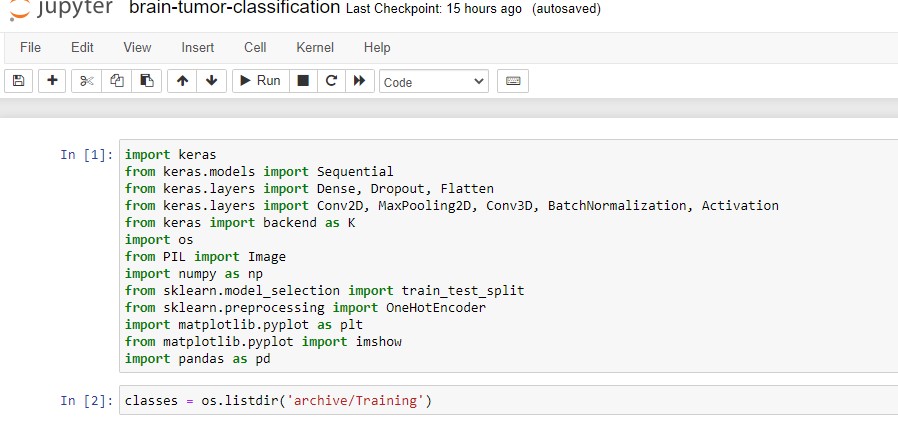
## 5.2 Testing

The major objective of testing to check it is depended on all platforms and to check the compatibility. All projects should be tested prior to the working of application

## 5.3 Types of Testing

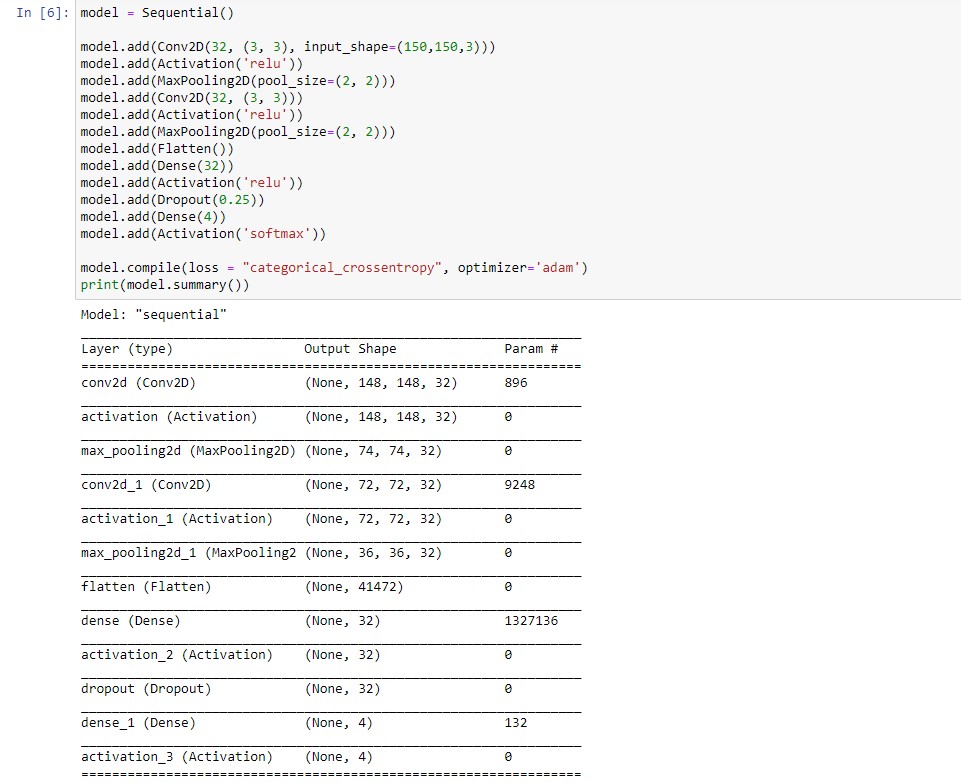
### 5.3.1 Unit testing

Input



### 5.3.2 Integration testing

Input



### 5.3.3 Functional testing

This sort of testing is utilized to test the usefulness of item. It looks at the genuine and expected yield utilizing input. As Clustering is viewed as an unaided learning technique since we don’t have the ground truth to think about the yield of the bunching calculation to the genuine names to assess its presentation.

### 5.3.4 White Box Testing

Logics of each and every aspect are created by flow pictures and logics of all functions are tested in different conditions.

### 5.3.5 Black Box Testing

All functional needs of project are executed and creating several test cases to project. This testing is used to identify the faults in data structures ,performance and interface.

## 5.4 Testing Strategy

The following testing case will give a proper demonstration on how the data is classified and verified through the proposed system and it can lead to a perfect analysis to verify the set of data.

Chapter 6

# RESULTS AND DISCUSSIONS

## 6.1 Efficiency of the Proposed System

In this Neural Autoregressive Distribution Estimation (NADE) model present how deep NADE models can be trained to be agnostic to the ordering of input dimensions used by the autoregressive product rule decomposition.and also explains neural network architectures applied to the problem of unsupervised distribution andn density estimation.

## 6.2 Comparison of Existing and Proposed System

In existing system distribution estimation is calculated in only binary-valued observations.By using NADE we can achieve competitive performance in modeling both binary and real-valued observations.

## 6.3 Advantages of the Proposed System

The advantages of the convolutional neural network are the fact that it provides optimal accuracy of segmentation. However, this is at the cost of computational load image. With advances in computation, the implementation of convolutional neural networks and refinement of the structural segmentation of brain tumours can be enhanced

## 6.4 Sample Code

|  |
| --- |
| trainData = [ ] trainLabel = [ ] dim = (150 , 150) t r a i n P a t h = ” archive / Training ” index = 0 for dir in os . l i s t d i r ( t r a i n P a t h ) :  f i l e P a t h s = [ ] subDir = os . path . join ( trainPath , dir ) for f i l e in os . l i s t d i r ( subDir ) : imgFullPath = os . path . join ( subDir , f i l e ) f i l e P a t h s . append ( imgFullPath ) img = Image . open ( imgFullPath ) x = img . r e s i z e ( dim ) x = np . array ( x ) trainData . append ( np . array ( x ) ) trainLabel . append ( enc . transform ( [ [ index ] ] ) . toarray ( ) ) p r i n t ( names ( index ) ) p r i n t ( s t r ( dir ) ) index += 1  trainData = np . array ( trainData ) trainLabel = np . array ( trainLabel ) . reshape (2870 , 4) p r i n t ( trainData . shape ) p r i n t ( trainLabel . shape ) |

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Output



Fig 6.4: Output 1



Fig 6.5:Output 2

Chapter 7

# CONCLUSION AND FUTURE

ENHANCEMENTS

## 7.1 Conclusion

This model achieves high performance in front of a few imbalanced classification brain tumor datasets with high rate accuracy after trained by a 6-fold crossvalidation technique and Adam optimizer. This process removes undesired features and smooths the boundary of brain tumors and aso extract benificial features for image classification. As a result of this research, the hybrid structure is a beneficial tool that can be used in medical image processing applications.

## 7.2 Future Enhancements

Convolutional neural networks (CNNs) are a unique machine learning structure originally modelled on the human visual cortex [4]. The brain was studied due to the abundance of segmentation methods MRI creates pictures of soft tissue parts of the body that are sometimes hard to see using other imaging tests in MRI.

Chapter 8

# PLAGIARISM REPORT

Chapter 9

# SOURCE CODE & POSTER

PRESENTATION

## 9.1 Source code

|  |  |
| --- | --- |
| import keras from keras . models import Sequential from keras . layers import Dense , Dropout , F l a t t e n from keras . layers import Conv2D , MaxPooling2D , Conv3D ,  Activation  from keras import backend as K import os from PIL import Image import numpy as np from sklearn . model selection import t r a i n t e s t s p l i t from sklearn . preprocessing import OneHotEncoder import matplotlib . pyplot as p l t from matplotlib . pyplot import imshow import pandas as pd c l a s s e s = os . l i s t d i r ( ’ archive / Training ’ ) enc = OneHotEncoder ( ) enc . f i t ( [ [ 0 ] , [1] , [2] , [ 3 ] ] ) def names ( number ) :  i f ( number == 0) : return c l a s s e s [0] e l i f ( number == 1) : return c l a s s e s [1] | BatchNormalization , |

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22e l i f ( number == 2) : 23return c l a s s e s [2] 24e l i f ( number == 3) :

25return c l a s s e s [3]

26trainData = [ ]

27trainLabel = [ ]

28dim = (150 , 150)

29t r a i n P a t h = ” archive / Training ”

30index = 0

31for dir in os . l i s t d i r ( t r a i n P a t h ) :

32f i l e P a t h s = [ ]

33subDir = os . path . join ( trainPath , dir )

34for f i l e in os . l i s t d i r ( subDir ) :

35imgFullPath = os . path . join ( subDir , f i l e )

36f i l e P a t h s . append ( imgFullPath )

37img = Image . open ( imgFullPath )

38x = img . r e s i z e ( dim )

39x = np . array ( x )

40trainData . append ( np . array ( x ) )

41trainLabel . append ( enc . transform ( [ [ index ] ] ) . toarray ( ) )

42p r i n t ( names ( index ) )

43p r i n t ( s t r ( dir ) )

44index += 1

45

46trainData = np . array ( trainData )

47trainLabel = np . array ( trainLabel ) . reshape (2870 , 4)

48p r i n t ( trainData . shape )

49p r i n t ( trainLabel . shape )

50testData = [ ]

51testLabel = [ ]

52dim = (150 , 150)

53t e s t P a t h = ” archive / Testing ”

54index = 0

55for dir in os . l i s t d i r ( t e s t P a t h ) :

56f i l e P a t h s = [ ]

57subDir = os . path . join ( testPath , dir )

58for f i l e in os . l i s t d i r ( subDir ) :

59imgFullPath = os . path . join ( subDir , f i l e )

60f i l e P a t h s . append ( imgFullPath )

61img = Image . open ( imgFullPath )

62x = img . r e s i z e ( dim )

63x = np . array ( x )

64testData . append ( np . array ( x ) )

65testLabel . append ( enc . transform ( [ [ index ] ] ) . toarray ( ) )

66p r i n t ( names ( index ) )

67p r i n t ( s t r ( dir ) )

68index += 1

69testData = np . array ( testData )

70testLabel = np . array ( testLabel ) . reshape (394 , 4)

71p r i n t ( testData . shape )

72p r i n t ( testLabel . shape )

73model = Sequential ( )

74

75 model . add (Conv2D(32 , (3 , 3) , input shape =(150 ,150 ,3) ) )

76model . add ( Activation ( ’ relu ’ ) )

77 model . add ( MaxPooling2D ( pool size =(2 , 2) ) )

78model . add (Conv2D(32 , (3 , 3) ) )

79model . add ( Activation ( ’ relu ’ ) )

80 model . add ( MaxPooling2D ( pool size =(2 , 2) ) )

81model . add ( F l a t t e n ( ) )

82model . add ( Dense (32) )

83model . add ( Activation ( ’ relu ’ ) )

84model . add ( Dropout (0.25) )

85model . add ( Dense (4) )

86model . add ( Activation ( ’ softmax ’ ) )

87

88 model . compile ( loss = ” c a t e g o r i c a l c r o s s e n t r o p y ” , optimizer = ’adam ’ )

89p r i n t ( model . summary ( ) )

90# Plot t r a i n i n g & v a l i d a t i o n loss values

91p l t . plot ( h i s t o r y . h i s t o r y [ ’ loss ’ ] )

92 p l t . plot ( h i s t o r y . h i s t o r y [ ’ v a l l o s s ’ ] )

93p l t . t i t l e ( ’Model Loss ’ )

94p l t . ylabel ( ’ Loss ’ )

95p l t . xlabel ( ’Epoch ’ )

96p l t . legend ( [ ’ Test ’ , ’ Validation ’ ] , loc= ’ upper r i g h t ’ )

97p l t . show ( )

98 img = Image . open ( ’ archive / Testing / glioma tumor / image (10) . jpg ’ )

99x = np . array ( img . r e s i z e ( dim ) )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x = x . reshape (1 ,150 ,150 ,3) answ = model . predict on batch ( x ) c l a s s i f i c a t i o n = np . where ( answ == np . amax ( answ ) ) [ 1 ] [ 0 ] imshow ( img ) p r i n t ( s t r ( answ [ 0 ] [ c l a s s i f i c a t i o n ]\*100) + ’% Confidence c l a s s i f i c a t i o n ) ) | This | Is | ’ | + names ( |

100

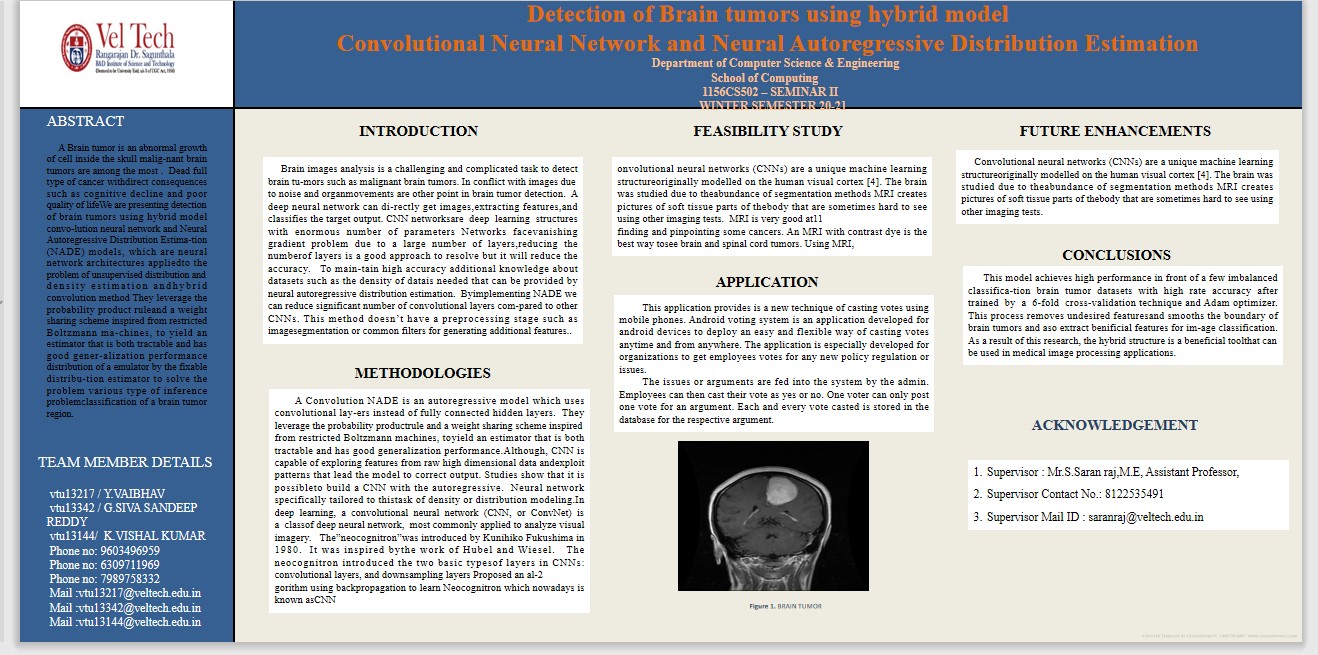
101

102

103

104

## 9.2 Poster Presentation



# References

1. Raheleh Hashemzehia , neural networking in cnn Received 23 October 2019Accepted 2 June 2020Available online 10 June 2020
2. Seyyed Jauad Seyyed Mahdaui,

Development of deep learning algorithm using convolutional neural network for medical imaging. Int] Eng Adv Technol 2020;

1. Rav`ı D, Wong C,

Lo B, et al. Deep learning for health informatics. IEEE J Biomed Health Inf

2017;

1. ] Litjens G, survey on deep learning in medical image analysis. Med Image Anal ,IEEE ,09th march 2018
2. S. Somasundaram

Machine Learning, Big Data, and Parallel Computing (Com-IT-Con),IEEE ,India, 16th Feb 2019