

BRAIN TUMOR DETECTION USING DEEP LEARNING

*A project report submitted in partial fulfillment of the requirements for the
award for the degree of*

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

B.Goutami-318126512065

D.Harshavardhan-318126512074

D. Harshith Chowdhary-318126512075

P.Ganesh-319126512L08

Under the guidance of

Mr. A. SIVA KUMAR

Assistant Professor



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC
with 'A Grade')*

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

(2021-2022)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)


*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with
'A' Grade)*

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

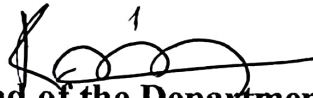


CERTIFICATE

This is to certify that the project report entitled "BRAIN TUMOR DETECTION USING DEEP LEARNING" submitted by B. Goutami (318126512065), D. Harshavardhan (318126512074), D. Harshith Chowdhary (318126512075), P. Ganesh (319126512L08) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.


Project Guide
Mr. A. Siva Kumar
M.Tech,(Ph.D)
Assistant Professor
Department of E.C.E.
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162


Head of the Department
Dr. V. Rajyalakshmi
Professor & HOD
Department of E.C.E
ANITS

Head of the Department
Department of ECE
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa-531 162

ACKNOWLEDGMENT

We would like to express our deep gratitude to our project guide **Mr. A. Siva Kumar**, M. Tech, PhD, Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa**, for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of the Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank all **non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

Project Students

B. Goutami (318126512065)

D. Harshavardhan (318126512074)

D. Harshith Chowdhary (318126512075)

P. Ganesh (319126512L08)

ABSTRACT

Brain is one of the most complex organs in the human body that works with billions of cells. A brain tumor arises when there is uncontrolled division of cells forming an abnormal group of cells around or inside the brain. That group of cells can affect the normal functionality of the brain activity and destroy the healthy cells. Brain tumor is of two types: Benign (non-cancerous) and Malignant (cancerous). Brain tumor detection and classification is the most troublesome and tedious task within the space of medicative image getting ready. Computer vision is playing important role in the field of human health care. This role is growing day by day. Various imaging methods like CT Scans, X-Ray, and MRI are available but MRI is the most reliable and safe. Currently, doctors locate the position and the area of brain tumor by looking at the MR Images of the brain of the patient manually. In this project, we are using deep learning with CNN architecture to accurately identify the tumor. We train, test and validate the deep learning model from the dataset. We used various convolutional layers in our project to make the prediction easy. Convolutional Neural Networks (CNN) is one of the most effective techniques in deep learning. CNN finds its applications in almost every field of research. Deep Learning structure extends the traditional neural networks by adding more hidden layers between the input and output. In this project, we used VGG16 to get remarkable accuracy on ImageNet dataset. We used Visual Studio Code to write the code and imported various modules like Keras, TensorFlow etc.

CONTENTS

LIST OF FIGURES	7
LIST OF TABLES	8
LIST OF ABBREVIATIONS	9
CHAPTER 1 INTRODUCTION	10
1.1 Project Objective	10
1.2 Project Outline	10
CHAPTER 2 LITERATURE SURVEY	11
CHAPTER 3 THEORITICAL ASPECTS	12
3.1 Brain Tumor	12
3.1.1 Causes of Tumor	15
3.1.2 Symptoms	17
3.1.3 Effects of Tumor	17
3.1.4 Diagnosis	18
3.2 Magnetic Resonance Imaging (MRI)	19
3.3 Image processing	20
3.3.1 Introduction	20
3.3.2 Fundamental steps involved in Image processing	20
3.3.3 Applications	22
3.3.4 Types of Image Processing	23
3.3.5 Benefits of Image Processing	24
3.3.6 Types of images	25
3.3.7 Image processing techniques	29
3.4 Noises in image	30
3.4.1 Types	30
3.5 Filtering	33
CHAPTER 4 DEEP LEARNING	35
4.1 Introduction	35
4.2 Importance of deep learning in today's world	35
4.3 Working of deep learning models	36

4.4 Convolutional Neural Network	36
4.4.1 Introduction	36
4.4.2 Working of CNN	37
4.4.3 Various layers and their importance	37
4.4.4 Applications	40
4.5 Applications of Deep Learning	41
CHAPTER 5 METHODOLOGY	44
5.1 Building a Deep Learning Model	46
5.2 Reading the training data	47
5.3 Split the dataset	48
5.4 Building the model	49
5.5 Compiling the model	51
5.6 Training the model	52
5.7 Predicting the data	53
5.8 Model capacity	53
5.9 Classification of model	54
CHAPTER 6 RESULTS AND FUTURE SCOPE	56
6.1 System Configuration	56
6.1.1 Software Requirements	58
6.1.2 Hardware Configuration	56
6.2 Scope for future	62
CHAPTER 7 CONCLUSION	63
REFERENCES	64

LIST OF FIGURES

Fig 3.3.3 1. Traffic Sensing Technologies	22
Fig 3.3.3 2. Face Detection	23
Fig 3.3.6 1. Binary image	25
Fig 3.3.6 2. Grey-scale image	26
Fig 3.3.6 3. Color image	26
Fig 3.3.6 4. 8-bit color format	27
Fig 3.3.6 5. 16-bit color format	27
Fig 3.3.6 6. 24-bit color format	27
Fig 3.3.7 1. Image Restoration	28
Fig 3.3.7 2. Linear filtering	29
Fig 4.4.1 CNN	34
Fig 4.4.2 Convolution in CNN	36
Fig 5.1 Deep Learning Model	43
Fig 5.4 ReLU Function	48
Fig 6.1.2 1. Image 1	60
Fig 6.1.2 2. Image 2	60
Fig 6.1.2 3. Image 3	61
Fig 6.1.2 4. Image 4	61
Fig 6.1.2 5. Brain image with no tumor	62
Fig 6.1.2 6. Brain image with tumor (detected)	62

LIST OF TABLES

Table 5.2 Reading the training data	44
Table 5.3 1. Split the dataset	45
Table 5.3 2. Data representing wage per hour	45

LIST OF ABBREVIATIONS

CT	Computed Tomography
MRI	Magnetic Resonance Imaging
RF	Radio Frequency
VGG 16	Visual Geometry Group 16
CNN	Convolutional Neural Network
DL	Deep Learning
CNS	Central Nervous System
EBV	Epstein – Barr virus
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ANN	Artificial Neural Network

CHAPTER 1

INTRODUCTION

1.1 Project Objective

Our study deals with automated brain tumor detection and classification. Normally the anatomy of the brain is analyzed by MRI scans or CT scans. The aim of the paper is tumor identification in brain MR images. The main reason for detection of brain tumors is to provide aid to clinical diagnosis. The aim is to provide an algorithm that guarantees the presence of a tumor by combining several procedures to provide a foolproof method of tumor detection in MR brain images.

1.2 Project Outline

The focus of this project is MR brain images tumor extraction and its representation in simpler form such that it is understandable by everyone. The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. The aim of this work is to define an algorithm that will result in extracted image of the tumor from the MR brain image. Finally, we detect whether the given MR brain image has tumor or not using Convolution Neural Network.

CHAPTER 2

LITERATURE SURVEY

In Medical diagnosis, robustness and accuracy of the prediction algorithms are very important, because the result is crucial for treatment of patients. There are many popular classification and clustering algorithms used for prediction. The goal of clustering a medical image is to simplify the representation of an image into a meaningful image and make it easier to analyze. Several Clustering and Classification algorithms are aimed at enhancing the prediction accuracy of diagnosis process in detecting abnormalities.

- K. Sudharani, T. C. Sarma and K. Satya Rasad, "Intelligent Brain Tumor lesion classification and identification from MRI images using a K-NN technique," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 777-780. DOI: 10.1109/ICCICCT.2015.7475384

K. Sudharani et al. [5] presented a K- nearest neighbor algorithm to the MR images to identify and confine the hysterically full-fledged part within the abnormal tissues. The proposed work is a sluggish methodology but produces exquisite effects. The accuracy relies upon the sample training phase.

- Li, Shutao, JT-Y. Kwok, IW-H. Tsang and Yaonan Wang. "Fusing images with different focuses using support vector machines." IEEE Transactions on neural networks 15, no. 6 (2004): 1555-1561.

J.T. Kwok et al. [7] delivered wavelet-based photograph fusion to easily cognizance at the object with all focal lengths as several vision-related processing tasks can be carried out more effortlessly when wholly substances within the images are bright. In their work Kwok et al. investigated with different datasets, and results show that presented work is extra correct as it does not get suffering from evenness at different activity stages computations.

CHAPTER 3

THEORITICAL ASPECTS

3.1 Brain Tumor

A brain tumor is a collection, or mass, of abnormal cells in your brain. Your skull, which encloses your brain, is very rigid. Any growth inside such a restricted space can cause problems. Brain tumors can be cancerous (malignant) or noncancerous (benign). When benign or malignant tumors grow, they can cause the pressure inside your skull to increase. This can cause brain damage, and it can be life-threatening.

Brain tumors are categorized as primary or secondary:

- A primary brain tumor originates in your brain. Many primary brain tumors are benign.
- A secondary brain tumor, also known as a metastatic brain tumor, occurs when cancer cells spread to your brain from another organ, such as your lung or breast.

Benign v/s Malignant Tumors:

Though benign brain tumors can cause many serious issues, they are not cancerous, meaning that they grow slowly and don't typically spread to other tissues.

They also usually have more clearly defined borders, making them easier to remove surgically, and they don't usually come back after removal.

On the other hand, malignant brain tumors are cancerous, grow rapidly, and can spread to other parts of your brain or central nervous system, which can cause life-threatening complications.

Types of brain tumors:

1. Primary Brain tumor

Primary brain tumors originate in your brain. They can develop from your:

- brain cells
- the membranes that surround your brain, which are called meninges
- nerve cells
- glands, such as the pituitary or pineal

Primary tumors can be benign or cancerous. In adults, the most common types of brain tumors are gliomas and meningiomas.

2. Gliomas

Gliomas are tumors that develop from glial cells. These cells normally:

- support the structure of your central nervous system
- provide nutrition to your central nervous system
- clean cellular waste
- break down dead neurons

Gliomas can develop from different types of glial cells.

The types of tumors that begin in glial cells include:

- astrocytic tumors, such as astrocytomas, which originate in the cerebrum
- oligodendroglial tumors, which are often found in the frontal temporal lobes

- glioblastomas which originate in the supportive brain tissue and are the most aggressive type

3. Other primary brain tumors

Other primary brain tumors include:

- Pituitary tumors which are usually benign
- pineal gland tumors, which can be benign or malignant
- ependymomas, which are usually benign
- craniopharyngiomas, which occur mostly in children and are benign but can have clinical symptoms like changes in vision and premature puberty
- primary central nervous system (CNS) lymphomas, which are malignant
- primary germ cell tumors of the brain, which can be benign or malignant
- meningiomas, which originate in the meninges
- schwannomas which originate in cells that produce the protective cover of your nerves (myelin sheath) called Schwann cells

Meningiomas are diagnosed Trusted Source more often in women than men, according to a study that grouped participants into men and women.

Schwannomas occur equally in both men and women. These tumors are usually benign, but they can cause complications because of their size and location. Cancerous meningiomas and schwannomas are rare but can be very aggressive.

4. Secondary brain tumors

Secondary brain tumors make up the majority of brain cancers. They start in one part of the body and spread, or metastasize, to the brain. The following can metastasize to the brain:

- lung cancer
- breast cancer
- kidney cancer
- skin cancer

Secondary brain tumors are always malignant. Benign tumors don't spread from one part of your body to another.

3.1.1 Causes of tumor

Most of the time, the cause of a brain tumor is unknown, but the following factors may raise a person's risk of developing a brain tumor:

- **Age.** Brain tumors are more common in children and older adults, although people of any age can develop a brain tumor.
- **Gender.** In general, men are more likely than women to develop a brain tumor. However, some specific types of brain tumors, such as meningioma, are more common in women.
- **Home and work exposures.** Exposure to solvents, pesticides, oil products, rubber, or vinyl chloride may increase the risk of developing a brain tumor. However, there is not yet scientific evidence that supports this possible link.
- **Family history.** About 5% of brain tumors may be linked to hereditary genetic factors or conditions, including Li-Fraumeni syndrome neurofibromatosis, nevoid basal cell carcinoma syndrome, tuberous sclerosis, Turcot syndrome, and von Hippel-Lindau disease. Scientists have also found "clusters" of brain tumors within

some families without a link to these known hereditary conditions. Studies are underway to try to find a cause for these clusters.

- **Exposure to infections, viruses, and allergens.** Infection with the Epstein-Barr virus (EBV) increases the risk of CNS lymphoma. EBV is more commonly known as the virus that causes mononucleosis or “mono.” In other research, high levels of a common virus called cytomegalovirus (CMV) have been found in brain tumor tissue. The meaning of this finding is being researched. Several types of other viruses have been shown to cause brain tumors in research on animals. More data are needed to find out if exposure to infections, other viruses, or allergens increase the risk of a brain tumor in people. Of note, studies have shown that patients with a history of allergies or skin conditions have a lower risk of glioma.
- **Electromagnetic fields.** Most studies evaluating the role of electromagnetic fields, such as energy from power lines or from cell phone use, show no link to an increased risk of developing a brain tumor in adults. Because of conflicting information regarding risk in children, the World Health Organization (WHO) recommends limiting cell phone use and promotes the use of a hands-free headset for both adults and children.
- **Race and ethnicity.** In the United States, white people are more likely to develop gliomas but less likely to develop meningioma than Black people. Also, people from northern Europe are more than twice as likely to develop a brain tumor as people in Japan.
- **Ionizing radiation.** Previous treatment to the brain or head with ionizing radiation, including x-rays, has been shown to be a risk factor for a brain tumor.
- **Head injury and seizures.** Serious head trauma has long been studied for its relationship to brain tumors. Some studies have shown a link between head trauma and meningioma but not between head trauma and glioma. A history of seizures has also been linked with brain tumors, but because a brain tumor can cause seizures, it is not known if seizures increase the risk of brain tumors, if seizures occur because of the tumor, or if anti-seizure medication increases the risk.

- **N-nitroso compounds.** Some studies of diet and vitamin supplementation seem to indicate that dietary N-nitroso compounds may raise the risk of both childhood and adult brain tumors. Dietary N-nitroso compounds are formed in the body from nitrites or nitrates found in some cured meats, cigarette smoke, and cosmetics. However, additional research is necessary before a definitive link can be established.

3.1.2 Symptoms

- Headaches
- Seizures or convulsions
- Difficulty thinking, speaking or finding words
- Personality or behavior changes
- Weakness, numbness or paralysis in one part or one side of the body
- Loss of balance, dizziness or unsteadiness
- Loss of hearing
- Vision changes
- Confusion and disorientation
- Memory loss

3.1.3 Effects of tumor

- Pressure or headache near the tumor.
- Loss of balance and difficulty with fine motor skills is linked with a tumor in the cerebellum.
- Changes in judgment, including loss of initiative, sluggishness, and muscle weakness or paralysis is associated with a tumor in the frontal lobe of the cerebrum.
- Partial or complete loss of vision is caused by a tumor in the occipital lobe or temporal lobe of the cerebrum.

- Changes in speech, hearing, memory, or emotional state, such as aggressiveness and problems understanding or retrieving words can develop from a tumor in the frontal and temporal lobe of the cerebrum.
- Altered perception of touch or pressure, arm or leg weakness on 1 side of the body, or confusion with left and right sides of the body are linked to a tumor in the frontal or parietal lobe of the cerebrum.
- Inability to look upward can be caused by a pineal gland tumor.
- Lactation, which is the secretion of breast milk, and altered menstrual periods, as well as growth in hands and feet during adulthood, are linked with a pituitary tumor.
- Difficulty swallowing, facial weakness or numbness, or double vision is a symptom of a tumor in the brain stem.
- Vision changes, including loss of part of the vision or double vision can be from a tumor in the temporal lobe, occipital lobe, or brain stem.

3.1.4 Diagnosis

In general, diagnosing a brain tumor usually begins with magnetic resonance imaging (MRI). Once MRI shows that there is a tumor in the brain, the most common way to determine the type of brain tumor is to look at the results from a sample of tissue after a biopsy or surgery. An MRI uses magnetic fields, not x-rays, to produce detailed images of the body. MRI can be used to measure the tumor's size. A special dye called a contrast medium is given before the scan to create a clearer picture. This dye can be injected into a patient's vein or given as a pill or liquid to swallow. MRIs create more detailed pictures than CT scans (see below) and are the preferred way to diagnose a brain tumor. A sample of the tumor's tissue is usually needed to make a final diagnosis. A biopsy is the removal of a small amount of tissue for examination under a microscope and is the only definitive way a brain tumor can be diagnosed. A pathologist then analyzes the sample(s). A pathologist is a doctor who

specializes in interpreting laboratory tests and evaluating cells, tissues, and organs to diagnose disease. A biopsy can be done as part of surgery to remove the entire tumor. Or surgery may be done as a separate procedure if completely removing the tumor is not possible because of its location or a patient's health. A CT scan takes pictures of the inside of the body using x-rays taken from different angles. A computer combines these pictures into a detailed, 3-dimensional image that shows any abnormalities or tumors. A CT scan can help find bleeding and enlargement of the fluid-filled spaces in the brain, called ventricles. Changes to bone in the skull can also be seen on a CT scan, and it can be used to measure a tumor's size. A CT scan may also be used if the patient cannot have an MRI, such as if the person has a pacemaker for their heart. Sometimes, a contrast medium is given before the scan to provide better detail on the image. This dye can be injected into a patient's vein or given as a pill or liquid to swallow.

3.2 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a medical imaging technique used in radiology to form pictures of the anatomy and the physiological processes of the body. MRI scanners use strong magnetic fields, magnetic field gradients, and radio waves to generate images of the organs in the body. MRI does not involve X-rays or the use of ionizing radiation, which distinguishes it from CT and PET scans. MRI is a medical application of nuclear magnetic resonance (NMR) which can also be used for imaging in other NMR applications, such as NMR spectroscopy. MRI is widely used in hospitals and clinics for medical diagnosis, staging and follow-up of disease. Compared to CT, MRI provides better contrast in images of soft-tissues, e.g. in the brain or abdomen. However, it may be perceived as less comfortable by patients, due to the usually longer and louder measurements with the subject in a long, confining tube. Additionally, implants and other non-removable metal in the body can pose a risk and may exclude some patients from undergoing an MRI examination safely.

MRI was originally called NMRI (nuclear magnetic resonance imaging), but "nuclear" was dropped to avoid negative associations. Certain atomic nuclei are able to absorb radio frequency energy when placed in an external magnetic field; the resultant evolving spin polarization can induce a RF signal in a radio frequency coil and thereby be detected.

3.3 Image Processing

3.3.1 Introduction

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analyzing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

3.3.2 Fundamental steps involved in image processing

Image Acquisition

Image acquisition is the first step in image processing. This step is also known as preprocessing in image processing. It involves retrieving the image from a source, usually a hardware-based source.

Image Enhancement

Image enhancement is the process of bringing out and highlighting certain features of interest in an image that has been obscured. This can involve changing the brightness, contrast, etc.

Image Restoration

Image restoration is the process of improving the appearance of an image. However, unlike image enhancement, image restoration is done using certain mathematical or probabilistic models.

Color Image Processing

Color image processing includes a number of color modeling techniques in a digital domain. This step has gained prominence due to the significant use of digital images over the internet.

Wavelets and Multiresolution Processing

Wavelets are used to represent images in various degrees of resolution. The images are subdivided into wavelets or smaller regions for data compression and for pyramidal representation.

Compression

Compression is a process used to reduce the storage required to save an image or the bandwidth required to transmit it. This is done particularly when the image is for use on the Internet.

Morphological Processing

Morphological processing is a set of processing operations for morphing images based on their shapes.

Segmentation

Segmentation is one of the most difficult steps of image processing. It involves partitioning an image into its constituent parts or objects.

Representation and Description

After an image is segmented into regions in the segmentation process, each region is represented and described in a form suitable for further computer processing. Representation deals with the image's characteristics and regional properties. Description deals with extracting quantitative information that helps differentiate one class of objects from the other.

Recognition

Recognition assigns a label to an object based on its description.

3.3.3 Applications

1. Medical Image Retrieval

Image processing has been extensively used in medical research and has enabled more efficient and accurate treatment plans. For example, it can be used for the early detection of breast cancer using a sophisticated nodule detection algorithm in breast scans.

2. Traffic Sensing Technologies

In the case of traffic sensors, we use a video image processing system or VIPS. This consists of a) an image capturing system b) a telecommunication system and c) an image processing system. When capturing video, a VIPS has several detection zones which output an “on” signal whenever a vehicle enters the zone, and then output an “off” signal whenever the vehicle exits the detection zone. These detection zones can be set up for multiple lanes and can be used to sense the traffic in a particular station.



Fig 3.3.3 1. Traffic sensing technologies

3. Image Reconstruction

Image processing can be used to recover and fill in the missing or corrupt parts of an image. This involves using image processing systems that have been trained extensively with existing photo datasets to create newer versions of old and damaged photos.

4. Face Detection

One of the most common applications of image processing that we use today is face detection. It follows deep learning algorithms where the machine is first trained with the specific features of human faces, such as the shape of the face, the distance between the eyes, etc. After teaching the machine these human face features, it will start to accept all objects in an image that resemble a human face. Face detection is a vital tool used in security, biometrics and even filters available on most social media apps these days.

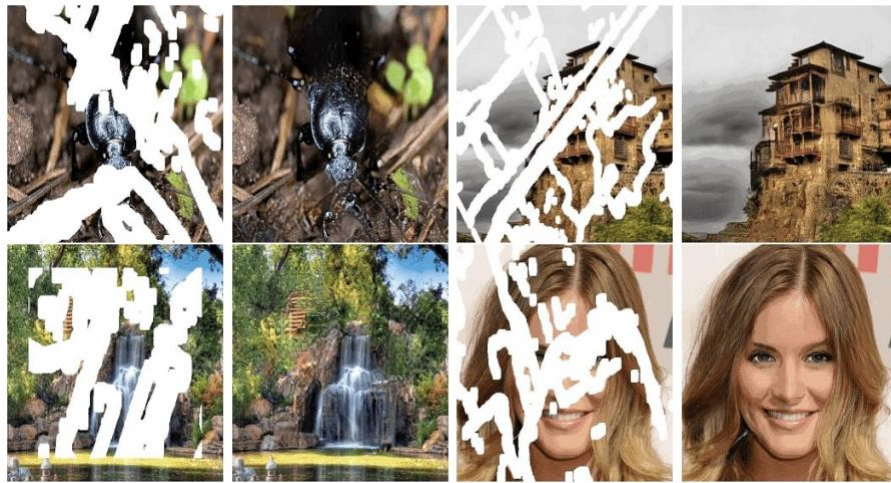


Fig 3.3.3 2. Face detection

3.3.4 Types of Image Processing

The two types of image processing used are analog image processing and digital image processing.

1. Analog image processing

In electrical engineering and computer science, analog image processing is any image processing task conducted on two-dimensional analog signals by analog means (as opposed to digital image processing). Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image

analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

2. Digital image processing

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

In this case, digital computers are used to process the image. The image will be converted into the digital form using a scanner – digitizer and then process it. It is defined as the subjecting numerical representation of objects to a series of operations in order to obtain the desired result. It starts with one image and produces a modified version of the image. It is therefore an image that takes one image into another. The term image processing generally refers to processing of a two-dimensional picture by a digital computer, in the broader context, it implies digital processing of a two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principle disadvantage of digital image processing is its versatility, repeatability and the preservation of original data precision.

3.3.5 Benefits of Image Processing

The implementation of image processing techniques has had a massive impact on many tech organizations. Here are some of the most useful benefits of image processing, regardless of the field of operation:

- The digital image can be made available in any desired format (improved image, X-Ray, photo negative, etc).
- It helps to improve images for human interpretation.
- Information can be processed and extracted from images for machine interpretation.
- The pixels in the image can be manipulated to any desired density and contrast.
- Images can be stored and retrieved easily.
- It allows for easy electronic transmission of images to third-party providers.

3.3.6 Types of images

There are three types of images. They are as following:

1. Binary Image

It is the simplest type of image. It takes only two values i.e, Black and White or 0 and 1. The binary image consists of a 1-bit image and it takes only 1 binary digit to represent a pixel. Binary images are mostly used for general shape or outline.

For Example: Optical Character Recognition (OCR).

Binary images are generated using threshold operation. When a pixel is above the threshold value, then it is turned white('1') and which are below the threshold value then they are turned black('0').

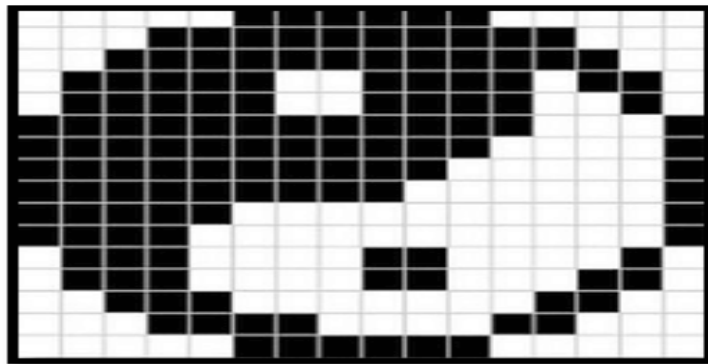


Fig. Binary image

Fig 3.3.6 1.Binary image

2. Grey-scale images

Grayscale images are monochrome images, Means they have only one color. Grayscale images do not contain any information about color. Each pixel determines available different grey levels.

A normal grayscale image contains 8 bits/pixel data, which has 256 different grey levels. In medical images and astronomy, 12 or 16 bits/pixel images are used.



Fig. Gray-scale image

Fig 3.3.6 2. Grey-scale image

3. Color images

Color images are three band monochrome images in which, each band contains a different color and the actual information is stored in the digital image. The color images contain gray level information in each spectral band.

The images are represented as red, green and blue (RGB images). And each color image has 24 bits/pixel means 8 bits for each of the three color band(RGB).

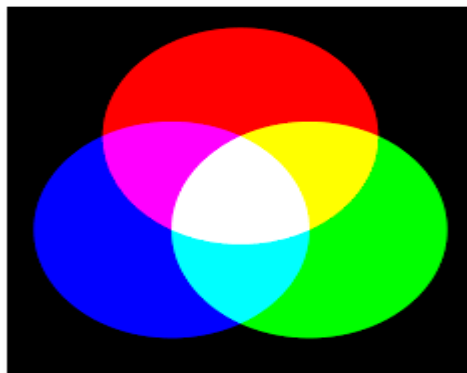


Fig. Colour image

Fig 3.3.6 3. Color image

8-bit color format

8-bit color is used for storing image information in a computer's memory or in a file of an image. In this format, each pixel represents one 8 bit byte. It has 0-255 range of colors, in which 0 is used for black, 255 for white and 127 for gray color. The 8-bit color format is also known as a grayscale image. Initially, it was used by the UNIX operating system.



Fig 3.3.6 4. 8-bit color format

16-bit color format

The 16-bit color format is also known as high color format. It has 65,536 different color shades. It is used in the system developed by Microsoft. The 16-bit color format is further divided into three formats which are Red, Green, and Blue also known as RGB format.

In RGB format, there are 5 bits for R, 6 bits for G, and 5 bits for B. One additional bit is added in green because in all the 3 colors green color is soothing to eyes.

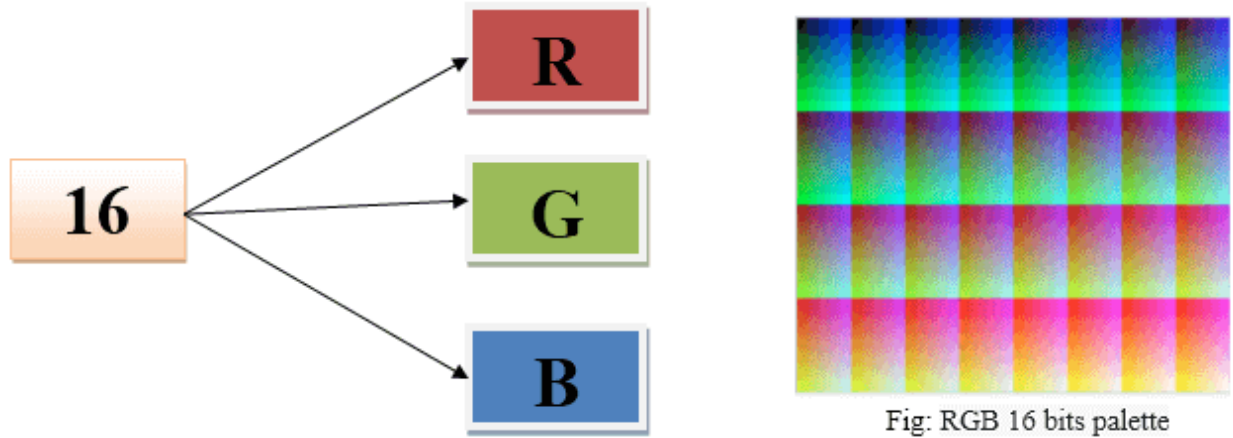


Fig: RGB 16 bits palette

Fig 3.3.6 5. 16-bit color format

24-bit color format

The 24-bit color format is also known as the true color format. The 24-bit color format is also distributed in Red, Green, and Blue. As 24 can be equally divided on 8, so it is distributed equally between 3 different colors like 8 bits for R, 8 bits for G and 8 bits for B.

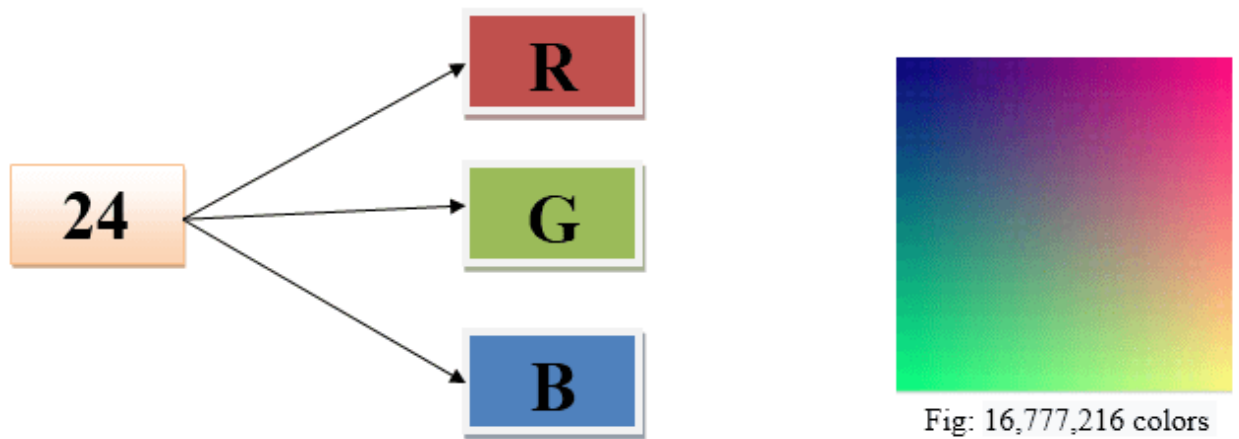


Fig: 16,777,216 colors

Fig 3.3.6 6. 24-bit color format

3.3.7 Image processing techniques

1. Image Restoration

An image deteriorates for many reasons, for example, an old image of your grandparents which was taken with the old tech camera could become hazy or may lose its original form. This could happen if the image goes under some physical stress or if it's in digital form it could deteriorate by motion blur or additive noise. So how are you going to restore it? Maybe it wasn't possible 50 years back but now – it is. Researchers came up with a Degradation Model that can undo the deterioration effects on the input image. The degradation model works as a convolution with a linear shift-invariant. So we take an Image before the degradation which is called “True Image” and an Image after degradation which is called “Observed Image” with the degradation filter which estimates the “True Image”.

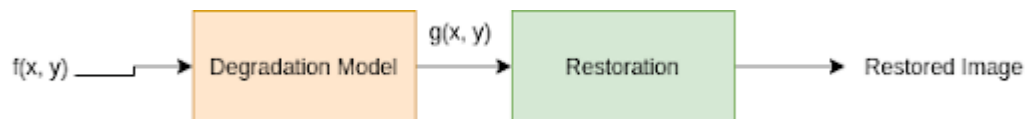


Fig 3.3.7 1. Image restoration

2. Linear Filtering

Linear filtering is a process in which the value of the output pixel is linear combinations of the neighboring input pixels. This process is done by a technique called Convolution.

Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel.

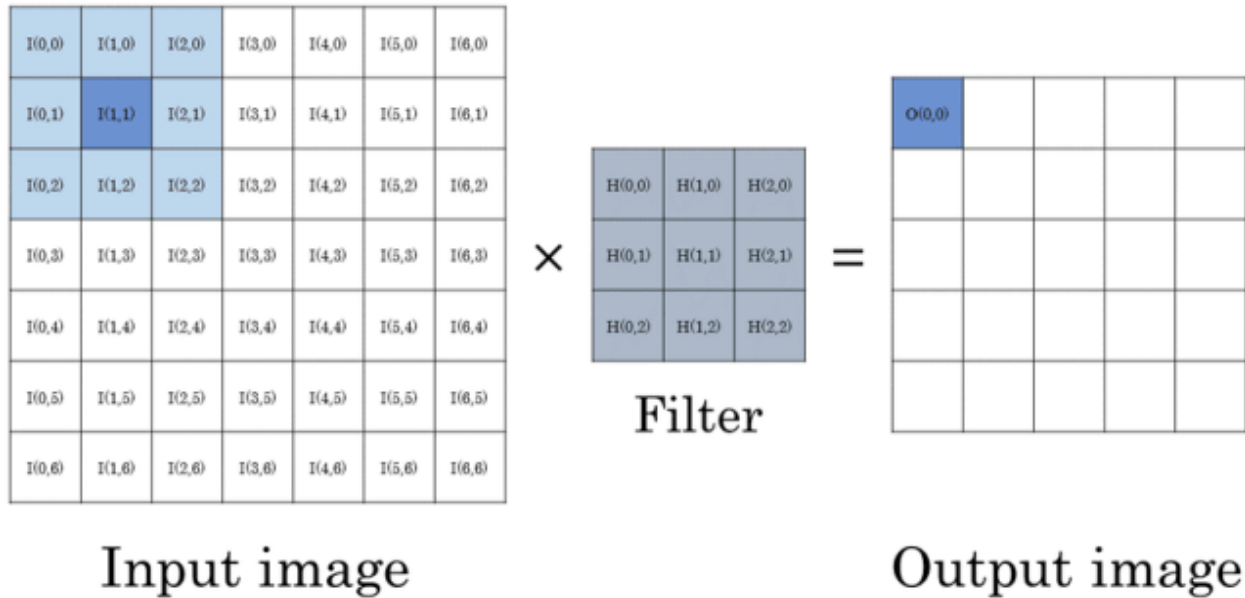


Fig 3.3.7 2. Linear filtering

We have an input image and a kernel with an anchor point. In the above diagram, it's $H(1, 1)$. This filter works as a sliding window to convolve over the image.

We multiply each pixel by the corresponding kernel and then take the sum. That sum becomes a new pixel in the output image.

3.4 Noises in image

3.4.1 Types

Gaussian Noise

Principal sources of Gaussian noise in digital images arise during acquisition. The sensor has inherent noise due to the level of illumination and its own temperature, and the electronic circuits connected to the sensor inject their own share of electronic circuit noise.

A typical model of image noise is Gaussian, additive, independent at each pixel, and independent of the signal intensity, caused primarily by Johnson–Nyquist noise (thermal noise), including that which comes from the reset noise of capacitors ("kTC noise"). Amplifier noise is a major part of the "read noise" of an image sensor, that is, of the constant noise level in dark areas of the image. In color cameras where more amplification is

used in the blue color channel than in the green or red channel, there can be more noise in the blue channel. At higher exposures, however, image sensor noise is dominated by shot noise, which is not Gaussian and not independent of signal intensity. Also, there are many Gaussian denoising algorithms.

Salt and Pepper Noise

Fat-tail distributed or "impulsive" noise is sometimes called salt-and-pepper noise or spike noise. An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission, etc. It can be mostly eliminated by using dark frame subtraction, median filtering, combined median and mean filtering and interpolating around dark/bright pixels. Dead pixels in an LCD monitor produce a similar, but non-random, display.

Shot Noise

The dominant noise in the brighter parts of an image from an image sensor is typically that caused by statistical quantum fluctuations, that is, variation in the number of photons sensed at a given exposure level. This noise is known as photon shot noise. Shot noise has a root-mean-square value proportional to the square root of the image intensity, and the noises at different pixels are independent of one another. Shot noise follows a Poisson distribution, which except at very high intensity levels approximates a Gaussian distribution.

In addition to photon shot noise, there can be additional shot noise from the dark leakage current in the image sensor; this noise is sometimes known as "dark shot noise" or "dark-current shot noise". Dark current is greatest at "hot pixels" within the image sensor. The variable dark charge of normal and hot pixels can be subtracted off (using "dark frame subtraction"), leaving only the shot noise, or random component, of the leakage. If dark-frame subtraction is not done, or if the exposure time is long enough that the hot pixel charge exceeds the linear charge capacity, the noise will be more than just shot noise, and hot pixels appear as salt-and-pepper noise.

Quantization Noise

The noise caused by quantizing the pixels of a sensed image to a number of discrete levels is known as quantization noise. It has an approximately uniform distribution. Though it can be signal dependent, it will be signal independent if other noise sources are big enough to cause dithering, or if dithering is explicitly applied.

Anisotropic Noise

Some noise sources show up with a significant orientation in images. For example, image sensors are sometimes subject to row noise or column noise.

Periodic Noise

A common source of periodic noise in an image is from electrical or electromechanical interference during the image capturing process. An image affected by periodic noise will look like a repeating pattern has been added on top of the original image. In the frequency domain this type of noise can be seen as discrete spikes. Significant reduction of this noise can be achieved by applying notch filters in the frequency domain. The following images illustrate an image affected by periodic noise, and the result of reducing the noise using frequency domain filtering. Note that the filtered image still has some noise on the borders. Further filtering could reduce this border noise, however it may also reduce some of the fine details in the image. The trade-off between noise reduction and preserving fine details is application specific. For example, if the fine details on the castle are not considered important, low pass filtering could be an appropriate option. If the fine details of the castle are considered important, a viable solution may be to crop off the border of the image entirely.

3.5 Filtering

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel.

Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

Image filtering is done to improve the quality of the image. For ex-smoothing an image reduces noise, blurred images can be rectified. There are broadly two types of algorithms linear and non-linear. Linear filter is achieved through convolution and Fourier multiplication whereas Non-linear filter cannot be achieved through any of these. Its output is not the linear function of its input thus, its result varies in a non intuitive manner. Here, the following image shows how the median filter enhances the images by reducing the noise and smoothing. In order to do further processing like image segmentation, edge detection etc. noise should be eliminated. Median filter is the most effective non-linear filtering algorithm to detect and remove salt and pepper noise. Median filter retains the edges of the while removing the noise that's why it is used most widely thus, apart from trampling the noise up to 5% to 60%, it also preserves the image details. The noisy pixels are evaluated and labelled as noisy pixels and the switching based median filter is applied to other pixels which are not noisy. Bilateral filter is a type of non-linear filter, it reduces noise by smoothing and preserves edges of the images. It takes weighted sum of the pixels which are nearby of each pixel and replaces the intensity of the pixels with the average of that weighted sum. Filtering of image is an important process done in image processing. It can be done for noise removal, blur removal, edge detection etc. Linear and non-linear filters are the algorithms which are used for filtering. Right filter should be selected for any specific purpose. If the image or input given has less amount of noise but the magnitude is high then non-linear filters are used whereas linear low-pass filter is sufficient when the input given contains noise in large amount but the magnitude of noise is low. Linear filters are the most frequently used filters as it is simplest and fastest. Unlike non-linear filters, the linear filtering is done through applying the algorithm on the

neighbour pixels of the input pixels in the image. The neighbourhood pixels are identified through their locations which are relative to the input pixel. Algorithms used for linear filtering are Box blur, Gaussian, bilateral and Hann window. In box blur an image with 9x9 pixel values, can be considered having a 3x3 neighbourhood values. smoothing is achieved by averaging the neighbourhood pixel values of the particular pixel in the output image.

CHAPTER 4

DEEP LEARNING

4.1 Introduction

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

4.2 Importance of deep learning in today's world

Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images.

While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful:

Deep learning requires large amounts of labeled data. For example, driverless car development requires millions of images and thousands of hours of video.

Deep learning requires substantial computing power. High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.

4.3 Working of deep learning models

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks.

The term “deep” usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150.

Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

4.4 Convolutional Neural Network

4.4.1 Introduction

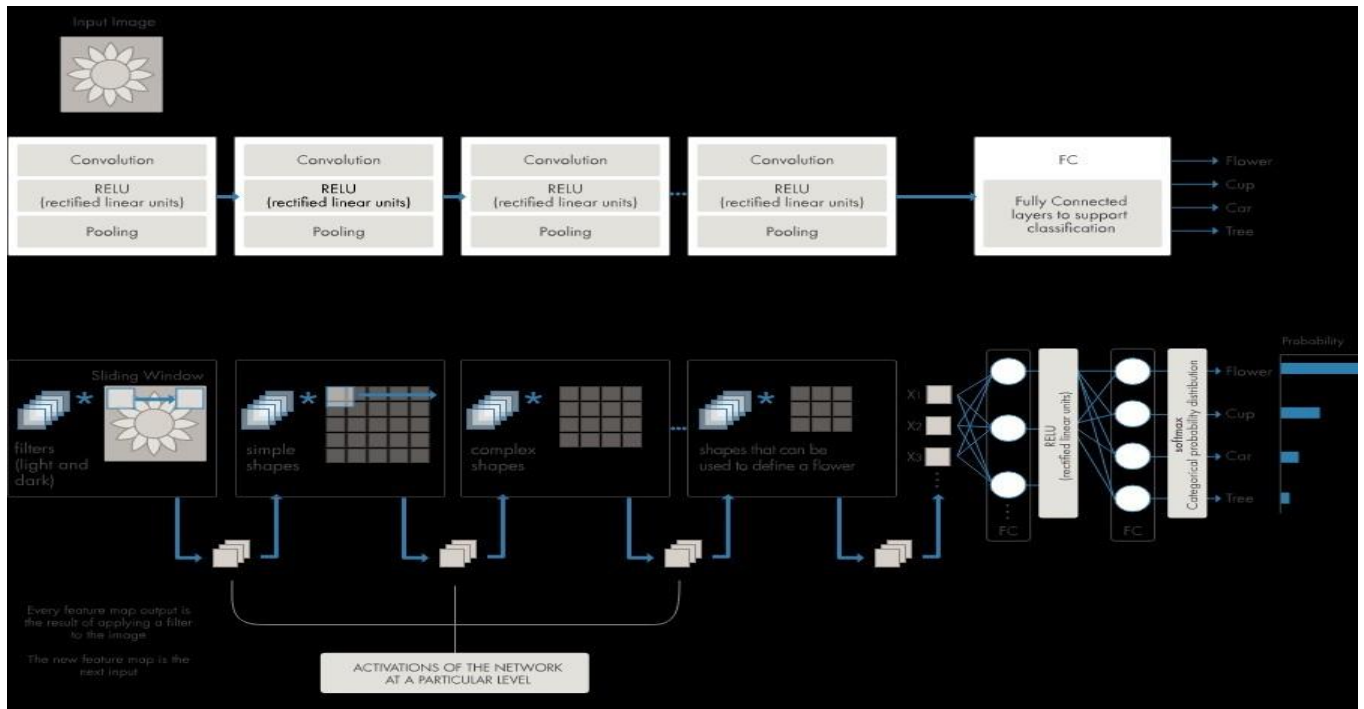


Fig 4.4.1 Convolutional Neural Network

One of the most popular types of deep neural networks is known as convolutional neural networks (CNN or ConvNet). The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution.

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery. CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation- equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain–computer interfaces, and financial time series.

4.4.2 Working of CNN

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

1. Convolutional layer
2. Pooling layer
3. Fully-connected (FC) layer

1. Convolutional Layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which

correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.

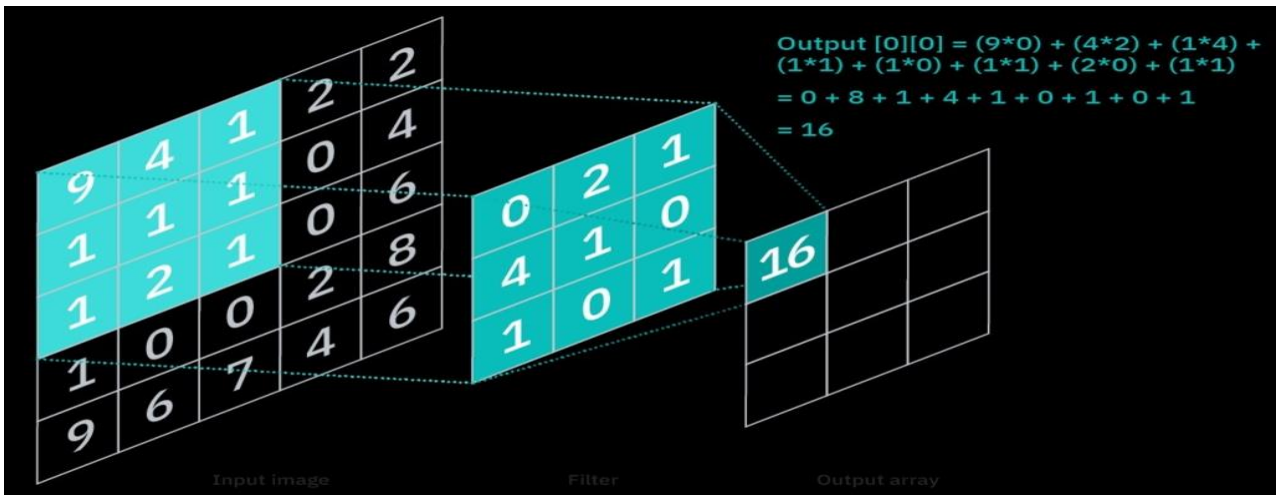


Fig 4.4.2 Convolution in CNN

As you can see in the image above, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as “partially connected” layers. However, this characteristic can also be described as local connectivity.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.
2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.
3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

Valid padding: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.

Same padding: This padding ensures that the output layer has the same size as the input layer

Full padding: This type of padding increases the size of the output by adding zeros to the border of the input.

Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

2. Pooling Layer

Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead,

the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

1. Max Pooling

As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

2. Average Pooling

As the filter moves across the input, it calculates the average value within the receptive field to send to the output array. While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

3. Fully – Connected Layer

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

4.4.4 Applications of CNN

1. Image recognition Edit

CNNs are often used in image recognition systems. In 2012 an error rate of 0.23% on the MNIST database was reported. When applied to facial recognition, CNNs achieved a large decrease in error rate.

2. Video analysis Edit

Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream.

3. Anomaly Detection Edit

A CNN with 1-D convolutions was used on time series in the frequency domain (spectral residual) by an unsupervised model to detect anomalies in the time domain.

4.5 Applications of Deep Learning

1. Automatic speech recognition

Large-scale automatic speech recognition is the first and most convincing successful case of deep learning. LSTM RNNs can learn "Very Deep Learning" tasks that involve multi-second intervals containing speech events separated by thousands of discrete time steps, where one time step corresponds to about 10 ms. LSTM with forget gates is competitive with traditional speech recognizers on certain tasks.

The initial success in speech recognition was based on small-scale recognition tasks based on TIMIT.

2. Image recognition

Deep learning-based image recognition has become "superhuman", producing more accurate results than human contestants. This first occurred in 2011 in recognition of traffic signs, and in 2014, with recognition of human faces.

3. Deep learning-trained vehicles now interpret 360° camera views. Another example is Facial Dymorphology Novel Analysis (FDNA) used to analyze cases of human malformation connected to a large database of genetic syndromes.

4. Visual art processing Edit

Closely related to the progress that has been made in image recognition is the increasing application of deep learning techniques to various visual art tasks. DNNs have proven themselves capable, for example, of identifying the style period of a given painting.

5. Neural Style Transfer

Capturing the style of a given artwork and applying it in a visually pleasing manner to an arbitrary photograph or video generating striking imagery based on random visual input fields.

6. Virtual Assistants

Virtual Assistants are cloud-based applications that understand natural language voice commands and complete tasks for the user. Amazon Alexa, Cortana, Siri, and Google Assistant are typical examples of virtual assistants. They need internet-connected devices to work with their full capabilities. Each time a command is fed to the assistant, they tend to provide a better user experience based on past experiences using Deep Learning algorithms.



Fig 4.5 Virtual Assistants

7. Chatbots

Chatbots can solve customer problems in seconds. A chatbot is an AI application to chat online via text or text-to-speech. It is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites, and instant messaging the client. It delivers automated responses to user inputs. It uses machine learning and deep learning algorithms to generate different types of reactions.

8. Healthcare

Deep Learning has found its application in the Healthcare sector. Computer-aided disease detection and computer-aided diagnosis have been possible using Deep Learning. It is widely used for medical research, drug discovery, and diagnosis of life-threatening diseases such as cancer and diabetic retinopathy through the process of medical imaging.

9. News aggregation and fake news detection

Deep Learning allows you to customize news depending on the readers' persona. You can aggregate and filter out news information as per social, geographical, and economic parameters and the individual preferences of a reader. Neural Networks help develop classifiers that can detect fake and biased news and remove it from your feed. They also warn you of possible privacy breaches.

10. Robotics

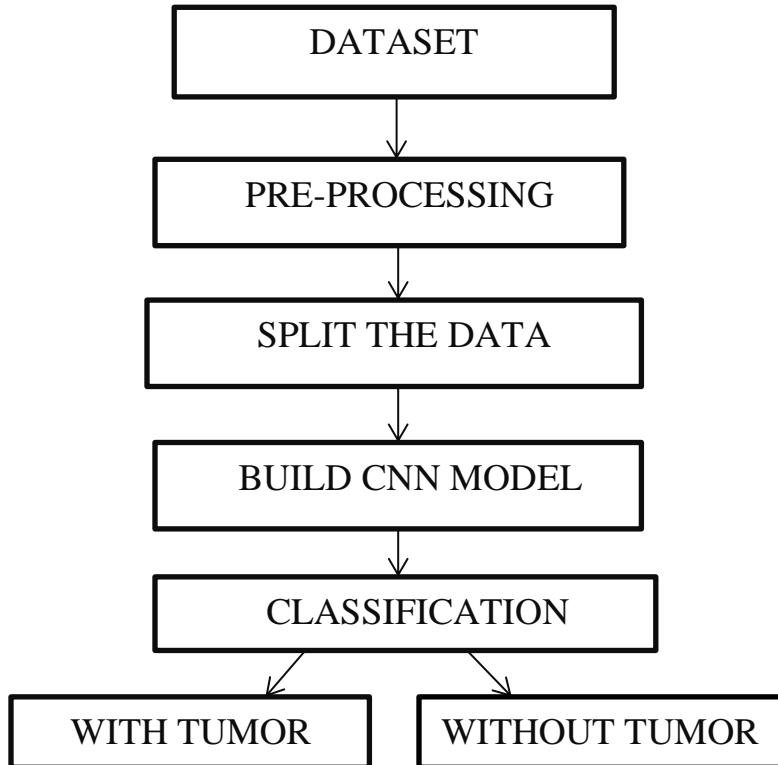
Deep Learning is heavily used for building robots to perform human-like tasks. Robots powered by Deep Learning use real-time updates to sense obstacles in their path and pre-plan their journey instantly. It can be used to carry goods in hospitals, factories, warehouses, inventory management, manufacturing products, etc.

Boston Dynamics robots react to people when someone pushes them around, they can unload a dishwasher, get up when they fall, and do other tasks as well.

CHAPTER 5

METHODOLOGY

FLOWCHART:



Dataset:

The dataset contains approximately around 3060-4000 MRI images of brain with tumor and without tumor. This dataset is downloaded from the Kaggle website. This dataset is downloaded from the Kaggle website. A dataset contains thousands of images in which we can train, test and evaluate the performance of deep learning algorithms. A dataset is made up of one or more database tables, each column representing a different variable and each row indicating a different record from the dataset.

Pre-processing:

Pre-processing is the process of modifying raw data before feeding it to a deep-learning system. It aids in the training process's acceleration. We can improve the image's quality and analyze it more thoroughly.

Some image pre-processing techniques are: Gray - scale conversion, Data augmentation, Normalization etc. Important operations like re-sizing and re-scaling are done in pre- processing. Preprocessing allows us to remove undesirable distortions and boost specific attributes that are relevant for the application we're working on. These features may vary based on the application.

Split the data:

When developing a deep-learning model, it is required to divide the dataset into three sections: train, test, and validate. This method is used to predict how well deep learning algorithms would perform. A dataset is divided into two subgroups using this method. The training dataset is the first collection of data to which the model is applied. The model receives the dataset's input element, which generates predictions and compares them to the predicted values. The model receives the dataset's input element, which produces predictions and compares them to the predicted values. The dataset's input element is given to the model, which then makes predictions and compares them to the predicted values. The second subset is not used to train the model; instead, the dataset's input element is given to the model, which then makes predictions and compares them to the predicted values. The test dataset is the second of the two datasets:

- Train Dataset: This is the data set to which the machine learning model is applied.
- A dataset is used to see how the machine learning model performs.

We must decide on percentage splits based on our needs. Some split percentages are:

- 80% for training, 20% for testing
- 67% for training, 33% for testing
- 50% for training, 50% for testing

5.1 Building a Deep Learning Model

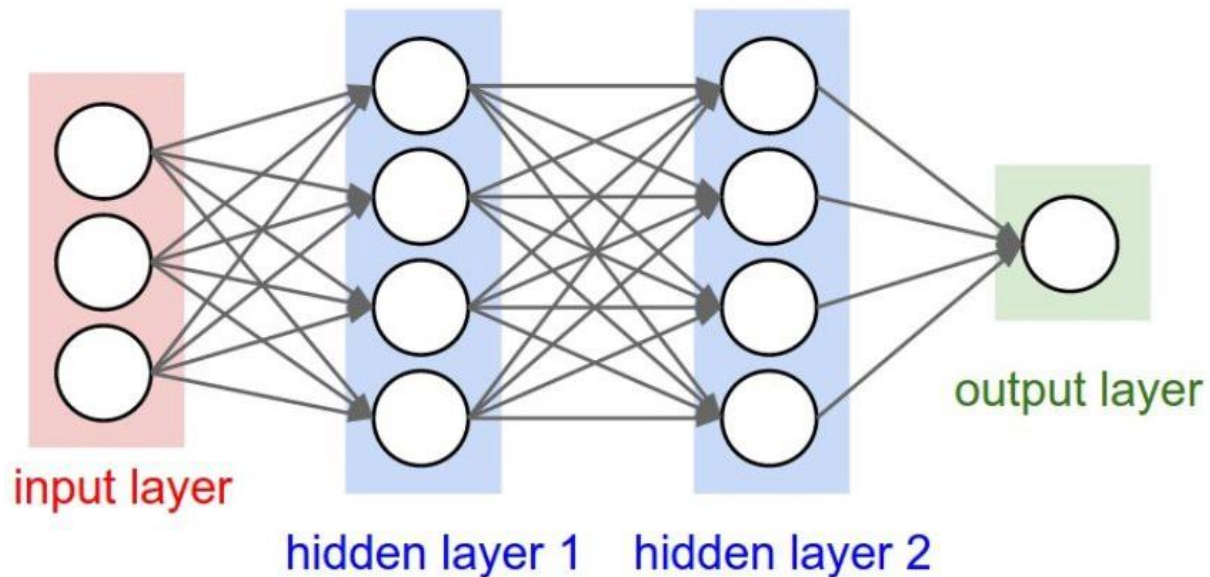


Fig 5.1 Deep Learning Model

Deep learning is an increasingly popular subset of machine learning. Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. The user does not need to specify what patterns to look for — the neural network learns on its own.

Keras is a user-friendly neural network library written in Python. In this tutorial, I will go over two deep learning models using Keras: one for regression and one for classification. We will build a regression model to predict an employee's wage per hour, and we will build a classification model to predict whether or not a patient has diabetes.

Note: The datasets we will be using are relatively clean, so we will not perform any data preprocessing in order to get our data ready for modeling. Datasets that you will use in future projects may not be so clean — for example, they may have missing values — so you may need to use data preprocessing techniques to alter your datasets to get more accurate results.

5.2 Reading the training data

For our regression deep learning model, the first step is to read in the data we will use as input. For this example, we are using the ‘hourly wages’ dataset. To start, we will use Pandas to read in the data.

‘df’ stands for dataframe. Pandas reads in the csv file as a dataframe. The ‘head()’ function will show the first 5 rows of the dataframe so you can check that the data has been read in properly and can take an initial look at how the data is structured.

Sample Code:

```
Import pandas as pd
#read in data using pandas
train_df = pd.read_csv('data/hourly_wages_data.csv')
#check data has been read in properly
train_df.head()
```

	wage_per_hour	union	education_yrs	experience_yrs	age	female	marr	south	manufacturing	construction
0	5.10	0	8	21	35	1	1	0	1	0
1	4.95	0	9	42	57	1	1	0	1	0
2	6.67	0	12	1	19	0	0	0	1	0
3	4.00	0	12	4	22	0	0	0	0	0
4	7.50	0	12	17	35	0	1	0	0	0

Table 5.2 Reading the training data

5.3 Split the dataset

Next, we need to split up our dataset into inputs (`train_X`) and our target (`train_y`). Our input will be every column except `'wage_per_hour'` because `'wage_per_hour'` is what we will be attempting to predict. Therefore, `'wage_per_hour'` will be our target.

We will use pandas `'drop'` function to drop the column `'wage_per_hour'` from our dataframe and store it in the variable `'train_X'`. This will be our input.

Sample Code:

```
#create a dataframe with all training data except the target column  
train_X = train_df.drop(columns=['wage_per_hour'])  
  
#check that the target variable has been removed  
train_X.head()
```

	union	education_yrs	experience_yrs	age	female	marr	south	manufacturing	construction
0	0	8	21	35	1	1	0	1	0
1	0	9	42	57	1	1	0	1	0
2	0	12	1	19	0	0	0	1	0
3	0	12	4	22	0	0	0	0	0
4	0	12	17	35	0	1	0	0	0

Table 5.3 1. Split the dataset

We will insert the column `'wage_per_hour'` into our target variable (`train_y`).

Sample Code:

```
#create a dataframe with only the target column  
train_y = train_df[['wage_per_hour']]
```



```
#view dataframe  
train_y.head()
```

	wage_per_hour
0	5.10
1	4.95
2	6.67
3	4.00
4	7.50

Table 5.3 2. Data representing wage per hour

5.4 Building the model

Next, we have to build the model. Here is the code:

Sample Code:

```
from keras.models import Sequential  
from keras.layers import Dense#create model  
model = Sequential()  
  
#get number of columns in training data  
n_cols = train_X.shape[1]  
  
#add model layers  
model.add(Dense(10, activation='relu', input_shape=(n_cols,)))  
model.add(Dense(10, activation='relu'))  
model.add(Dense(1))
```

The model type that we will be using is Sequential. Sequential is the easiest way to build a model in Keras. It allows you to build a model layer by layer. Each layer has weights that correspond to the layer the follows it.

We use the ‘add()’ function to add layers to our model. We will add two layers and an output layer.

‘Dense’ is the layer type. Dense is a standard layer type that works for most cases. In a dense layer, all nodes in the previous layer connect to the nodes in the current layer.

We have 10 nodes in each of our input layers. This number can also be in the hundreds or thousands. Increasing the number of nodes in each layer increases model capacity. I will go into further detail about the effects of increasing model capacity shortly.

‘Activation’ is the activation function for the layer. An activation function allows models to take into account nonlinear relationships. For example, if you are predicting diabetes in patients, going from age 10 to 11 is different than going from age 60–61.

The activation function we will be using is ReLU or Rectified Linear Activation. Although it is two linear pieces, it has been proven to work well in neural networks.

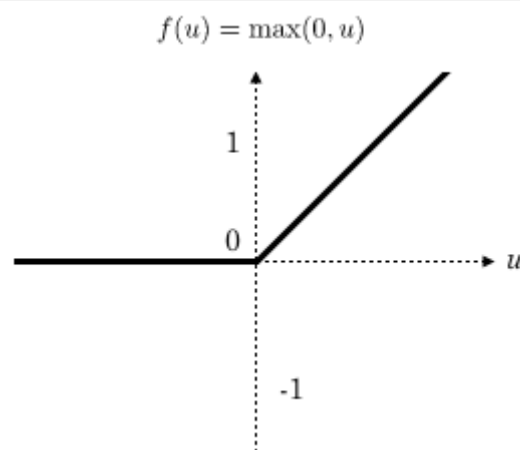


Fig 5.4 ReLU Function

The first layer needs an input shape. The input shape specifies the number of rows and columns in the input. The number of columns in our input is stored in 'n_cols'. There is nothing after the comma which indicates that there can be any amount of rows.

The last layer is the output layer. It only has one node, which is for our prediction.

5.5 Compiling the model

Next, we need to compile our model. Compiling the model takes two parameters: optimizer and loss.

The optimizer controls the learning rate. We will be using 'adam' as our optimizer. Adam is generally a good optimizer to use for many cases. The adam optimizer adjusts the learning rate throughout training.

The learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights (up to a certain point), but the time it takes to compute the weights will be longer.

For our loss function, we will use 'mean_squared_error'. It is calculated by taking the average squared difference between the predicted and actual values. It is a popular loss function for regression problems. The closer to 0 this is, the better the model performed.

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2$$

where N is the number of data points,
 f_i the value returned by the model and
 y_i the actual value for data point i .

Sample Code:

```
#compile model using mse as a measure of model performance  
model.compile(optimizer='adam', loss='mean_squared_error')
```

5.6 Training the model

Now we will train our model. To train, we will use the ‘fit()’ function on our model with the following five parameters: training data (train_X), target data (train_y), validation split, the number of epochs and callbacks.

The validation split will randomly split the data into use for training and testing. During training, we will be able to see the validation loss, which give the mean squared error of our model on the validation set. We will set the validation split at 0.2, which means that 20% of the training data we provide in the model will be set aside for testing model performance.

The number of epochs is the number of times the model will cycle through the data. The more epochs we run, the more the model will improve, up to a certain point. After that point, the model will stop improving during each epoch. In addition, the more epochs, the longer the model will take to run. To monitor this, we will use ‘early stopping’.

Early stopping will stop the model from training before the number of epochs is reached if the model stops improving. We will set our early stopping monitor to 3. This means that after 3 epochs in a row in which the model doesn’t improve, training will stop. Sometimes, the validation loss can stop improving then improve in the next epoch, but after 3 epochs in which the validation loss doesn’t improve, it usually won’t improve again.

Sample Code:

```
from keras.callbacks import EarlyStopping  
#set early stopping monitor so the model stops training when it won't improve anymore  
early_stopping_monitor = EarlyStopping(patience=3)  
#train model  
model.fit(train_X, train_y, validation_split=0.2, epochs=30,  
callbacks=[early_stopping_monitor])
```

5.7 Predicting the data

If you want to use this model to make predictions on new data, we would use the ‘predict()’ function, passing in our new data. The output would be ‘wage_per_hour’ predictions.

Sample code:

```
test_y_predictions = model.predict(test_X)
```

5.8 Model capacity

As you increase the number of nodes and layers in a model, the model capacity increases. Increasing model capacity can lead to a more accurate model, up to a certain point, at which the model will stop improving. Generally, the more training data you provide, the larger the model should be. We are only using a tiny amount of data, so our model is pretty small. The larger the model, the more computational capacity it requires and it will take longer to train.

Let’s create a new model using the same training data as our previous model. This time, we will add a layer and increase the nodes in each layer to 200. We will train the model to see if increasing the model capacity will improve our validation score.

Sample code:

```
#training a new model on the same data to show the effect of increasing model capacity

#create model
model_mc = Sequential()

#add model layers
model_mc.add(Dense(200, activation='relu', input_shape=(n_cols,)))
model_mc.add(Dense(200, activation='relu'))
model_mc.add(Dense(200, activation='relu'))
model_mc.add(Dense(1))

#compile model using mse as a measure of model performance
model_mc.compile(optimizer='adam', loss='mean_squared_error')#train model
model_mc.fit(train_X, train_y, validation_split=0.2, epochs=30,
callbacks=[early_stopping_monitor])
```

5.9 Classification of model

We are going to predict if patients have tumor or not.

Sample code:

```
#read in training data
train_df_2 = pd.read_csv('documents/data/tumor_data.csv')

#view data structure
train_df_2.head()

#create model
model_2 = Sequential()

#get number of columns in training data
n_cols_2 = train_X_2.shape[1]
```

```
#add layers to model  
model_2.add(Dense(250, activation='relu', input_shape=(n_cols_2,)))  
model_2.add(Dense(250, activation='relu'))  
model_2.add(Dense(250, activation='relu'))  
model_2.add(Dense(2, activation='softmax'))
```

The last layer of our model has 2 nodes — one for each option: the patient has diabetes or they don't.

The activation is 'softmax'. Softmax makes the output sum up to 1 so the output can be interpreted as probabilities. The model will then make its prediction based on which option has a higher probability.

```
#compile model using accuracy to measure model performance  
model_2.compile(optimizer='adam', loss='categorical_crossentropy',  
metrics=['accuracy'])
```

We will use 'categorical_crossentropy' for our loss function. This is the most common choice for classification. A lower score indicates that the model is performing better.

To make things even easier to interpret, we will use the 'accuracy' metric to see the accuracy score on the validation set at the end of each epoch.

```
#train model  
model_2.fit(X_2, target, epochs=30, validation_split=0.2,  
callbacks=[early_stopping_monitor])
```

CHAPTER 6

RESULTS AND FUTURE SCOPE

6.1 System Configuration

6.1.1 Software Requirements

Visual Studio Code, Python 3.9, PIP

Modules: Keras, Tensorflow, CV2, NumPy, Flask

Visual Studio Code:

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

Python:

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant Whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

PIP:

It is the package management system used to install and manage software packages written in Python.

NumPy:

NumPy is a general-purpose array-processing package. It provides a highperformance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Tensor Flow:

Tensor flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Keras:

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or Plaid ML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

OpenCV:

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by willow garage then Itseez (which was later acquired by Intel). The library is cross platform and free for use under the open source BSD license. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to

an ONNX model) and Caffe according to a defined list of supported layers. It promotes Open Vision Capsules, which is a portable format, compatible with all other formats.

6.1.2 Hardware Configuration

- Processor: Intel core i5 or above.
- 64-bit, quad-core, 2.5 GHz minimum per core
- Ram: 4 GB or more
- Hard disk: 10 GB of available space or more.
- Display: Dual XGA (1024 x 768) or higher resolution monitors
- Operating system: Windows

DATASET:

TUMOR IMAGES:



Fig 6.1.2 1. Image 1

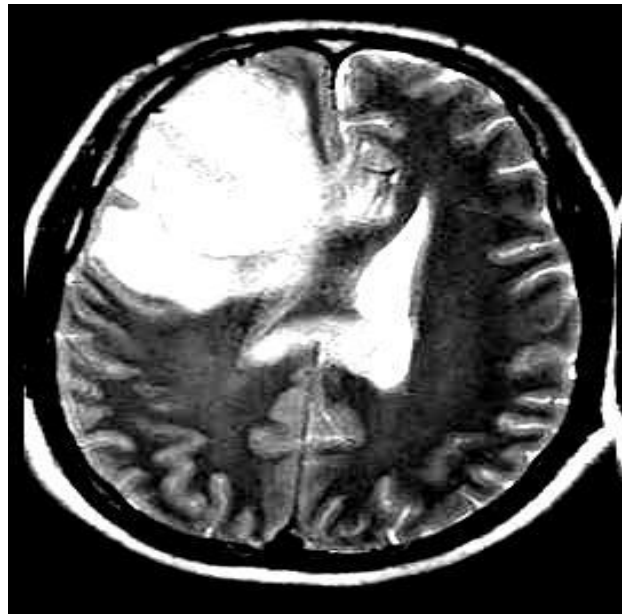


Fig 6.1.2 2. Image 2

NO TUMOR IMAGES:

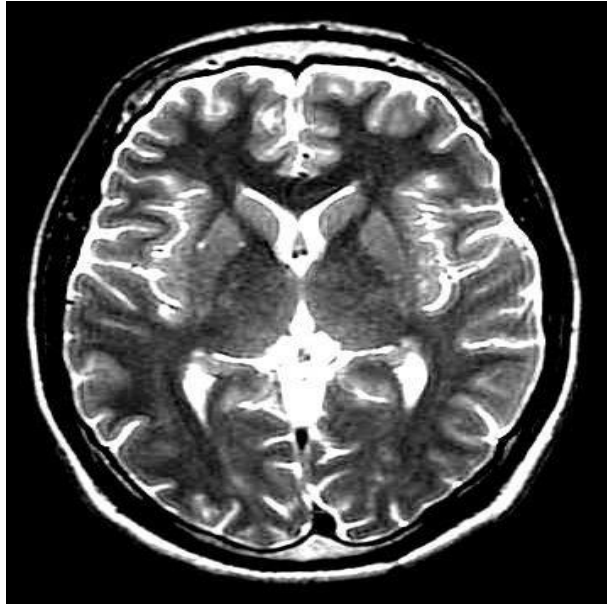


Fig 6.1.2 3. Image 3

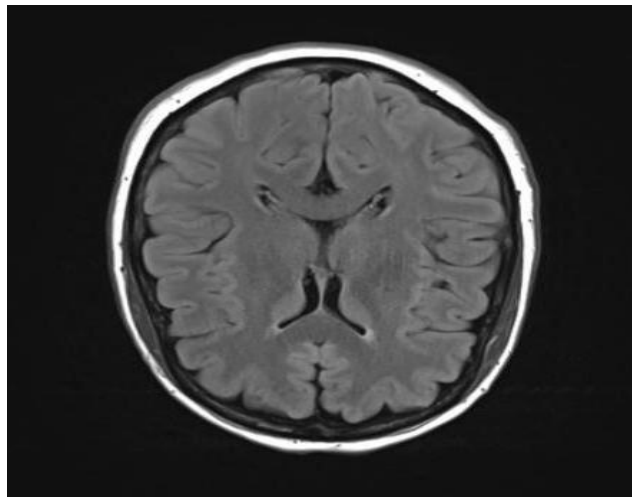


Fig 6.1.2 4. Image 4

PREDICTED OUTPUTS:

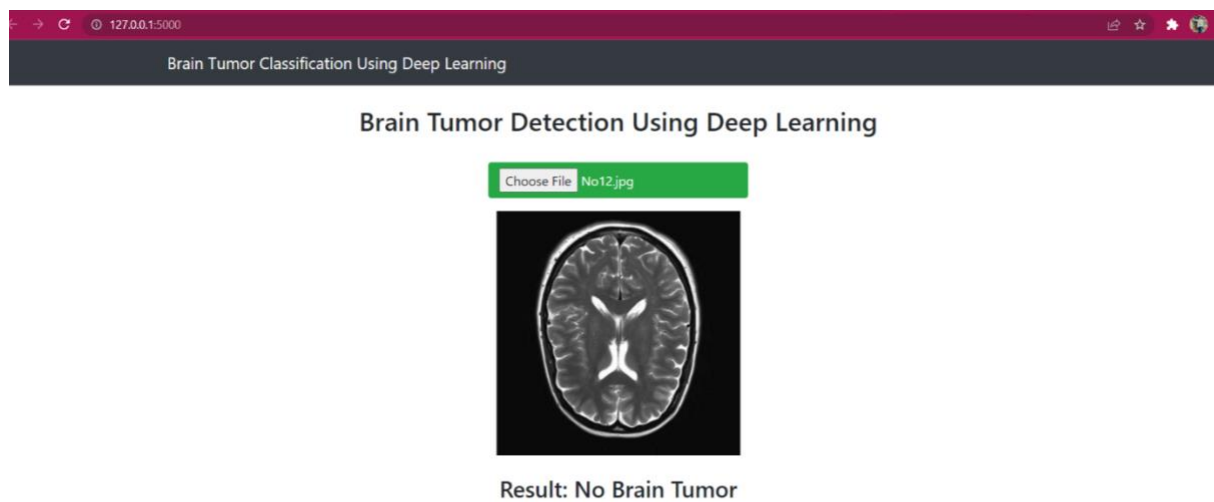


Fig 6.1.2 5. Brain image with no tumor

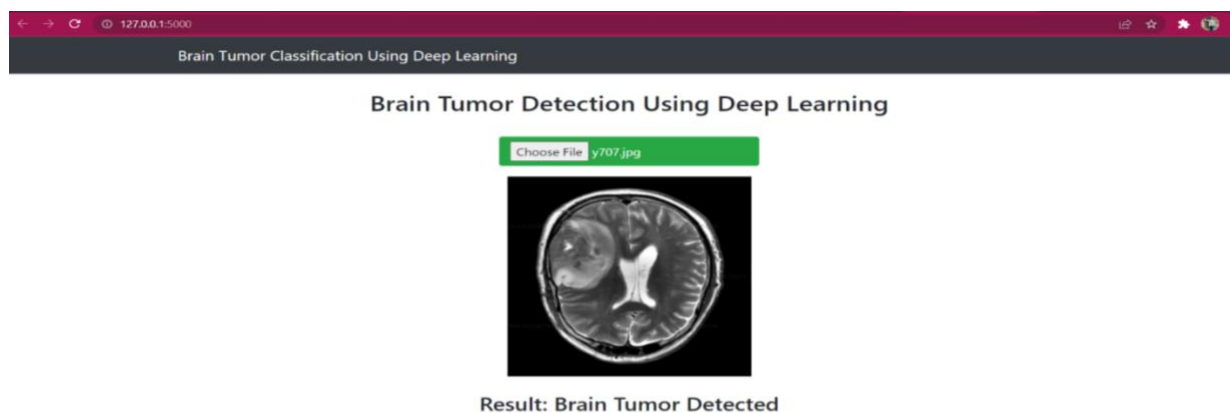


Fig 6.1.2 6. Brain image with tumor (detected)

6.2 Scope for future

It is observed on extermination that the proposed approach needs a vast training set for better accurate results; in the field of medical image processing, the gathering of medical data is a tedious job, and, in few cases, the datasets might not be available. In all such cases, the proposed algorithm must be robust enough for accurate recognition of tumor regions from MR Images. The proposed approach can be further improvised through in cooperating weakly trained algorithms that can identify the abnormalities with a minimum training data and also self-learning algorithms would aid in enhancing the accuracy of the algorithm and reduce the computational time.

CHAPTER 7

CONCLUSION

We proposed a computerized method for the identification of a brain tumor using the Convolution Neural Network. The input MR images are read from the local device using the file path and converted into grayscale images. These images are pre-processed using a filtering technique for the elimination of noises that are present inside the original image. The binary thresholding is applied to the denoised image, and Convolution Neural Network segmentation is applied, which helps in figuring out the tumor region in the MR images. The proposed model had obtained an accuracy of 96% and yields promising results without any errors and much less computational time. When this proposed methodology is applied to an abnormal brain MRI image, the tumor if any can be accurately detected. Hence, we can easily locate the tumor accurately.

REFERENCES

- [1]Litijens G, Kooi T, Bejnordi BE,Setio AA, Ciompi F, Ghafoorian M,et al, “A survey on deep learning in medical image analysis”. *Med Image Anal* 2017;42:60-88.
- [2]Anuse A, Vyas V, “A novel training algorithm for convolutional neural network”. *Contr Intell Syst* 2016;2(3):221-34.
- [3]Ravi D, Wong C, Deligianni F, Berthelot M, Andreu-Perez J, Lo B, et al “Deep Learning for health informatics”. *IEEE J Biomed Health Inf* 2017;21(1);4-21.
- [4]Das V, Rajan J.Techniques for “MRI brain tumor detection:a survey”,*Int J Res Comput Appl Inf Tech* 2016;4(3):53-6.
- [5] Shi, Z., He, L., Suzuki, K., Nakamura, T., & Itoh, H. (2009). “Survey on Neural Networks Used for Medical Image Processing”. *International Journal of computational science*, 3(1), 86–100.
- [6] S. Roy And S.K.Bandyopadhyay, “Detection And Qualification Of Brain Tumor From MRI Of Brain And Symmetric Analysis,” *International Journal Of Information And Communication Technology Research*, Volume 2 No.6, June 2012, Pp584-588.
- [7] Varuna Shree, N., Kumar, T.N.R. “Identification and classification of brain tumor MRI image’s with feature extraction using DWT and probabilistic neural network. *Brain Inf.* 5, 23–30 (2018) doi:10.1007/s40708-017-0075-5.
- [8] Pan, Yuehao & Huang, Weimin & Lin, Zhiping & Zhu, Wanzheng & Zhou, Jiayin & Wong, Jocelyn & Ding, Zhongxiang. (2015). “Brain tumor grading based on Neural Networks and Convolutional Neural Networks”. *Conference proceedings: 19 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference.* 2015. 699-702. 10.1109/EMBC.2015.7318458.

[9] P. Naga Srinivasu, T. Srinivasa Rao, Valentina Emilia Balas. (2020). “A systematic approach for identification of tumor regions in the human brain through HARIS algorithm, Deep Learning Techniques for Biomedical and Health Informatics”, Academic Press.Pages 97-118.

[10] Minz, Astina, and Chandrakant Mahobiya. “MR Image Classification Using Adaboost for Brain Tumor Type.” 2017 IEEE 7th International Advance Computing Conference (IACC) (2017): 701-705.