A Project report on

DIAGNOSIS OF BRAIN TUMOUR USING YOLO

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

K. Jaswanth Kumar RegdNo:19B91A0586

K. Tarun Kumar RegdNo:19B91A05C6

K. Sai Naga Vinay Varma RegdNo:19B91A05C4

K. Balaji Sai Rohith RegdNo:19B91A05C7

Under the esteemed Guidance of

B.N.V. NARASIMHA RAJU

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S.R.K.R ENGINEERING COLLEGE(A)

ChinnaAmiram, Bhimavaram, West Godavari Dist., A.P.

[2022-2023]

A Project report on

DIAGNOSIS OF BRAIN TUMOUR USING YOLO

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

K. Jaswanth Kumar	**RegdNo:19B91A0586**
K. Tarun Kumar	**RegdNo:19B91A05C6**
K. Sai Naga Vinay Varma	**RegdNo:19B91A05C4**
K. Balaji Sai Rohith	**RegdNo:19B91A05C7**

Under the esteemed Guidance of

B.N.V. NARASIMHA RAJU

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S.R.K.R ENGINEERING COLLEGE(A)

ChinnaAmiram, Bhimavaram, West Godavari Dist., A.P.

[2022-2023]

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

S.R.K.R ENGINEERING COLLEGE(A)

ChinnaAmiram, Bhimavaram, West Godavari Dist., A.P.

[2022-2023]

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "**Diagnosis of Brain Tumour using YOLO** " is the bonafide work of "**K. Jaswanth Kumar (19B91A0586), K. Tarun Kumar (19B91A05C6), K. Sai Naga Vinay Varma (19B91A05C4), K. Balaji Sai Rohith (19B91A05C7),**who carried out the project work under my supervision in partial fulfillment of the requirements for the award of thedegree of Bachelor of Technology in Computer Science and Engineering.

SUPERVISOR	**HEAD OF THE DEPARTMENT**
B. N. V. Narasimha Raju	Dr. V. Chandra Sekhar
(Assistant Professor)	(Professor)

SELF DECLARATION

We hereby declare that the project work entitled **“Diagnosis of Brain Tumour using YOLO”** is a genuine work carried out by us in B.Tech (Computer Science and Engineering) at SRKR Engineering College(A), Bhimavaram and has not been submitted either in part or full for the award of any other degree or diploma in any other institute or University.

1. K. Jaswanth Kumar	19B91A0586
2. K. Tarun Kumar	19B91A05C6
3. K. Sai Naga Vinay Varma	19B91A05C4
4. K. Balaji Sai Rohith	19B91A05C7

ABSTRACT

Detecting brain tumours early is crucial for patients' treatment and overall survival, as it is a rare and deadly disease with a poor prognosis. Neurologists and radiologists have a crucial role in this task, and computer-aided diagnosis systems utilizing magnetic resonance imaging (MRI) serve as valuable technology to support the diagnosis of brain tumours.

The process of manually detecting brain tumours from MRI scans can be challenging and imprecise. Hence, to tackle this issue, an approach combining a Convolutional Neural Network (CNN) for classification and the YOLOv4 (You Only Look Once version 4) object detection algorithm will be utilized for automatic detection of brain tumours.

The main objective is to save the lives of innocent people and decrease the mortality rate caused by brain tumours. To achieve this goal, a combination of the YOLOv4 (You Only Look Once version 4) object detection algorithm and a Convolutional Neural Network (CNN) for classification can be utilized to detect the presence of brain tumours accurately and efficiently.

TABLE OF CONTENTS

LIST OF FIGURES

1.INTRODUCTION

Brain tumours are a significant cause of death and disability globally as they invade the most vital organ in the human body. An abnormal growth of cells in the brain leads to the development of a brain tumour, making brain cancer the tenth leading cause of cancer-related deaths for both genders. Brain tumours can be primary or secondary. Primary brain tumours originate in the brain, while metastatic brain tumours (also called secondary brain tumours) are cancers that have spread to the brain from other body parts.

Brain tumours can be either benign or malignant. Benign tumours grow slowly and do not spread, while malignant tumours are more aggressive and require multiple treatments. Symptoms vary depending on tumour type, size, and location and may include headaches, seizures, limb weakness or numbness, difficulty with speech or comprehension, loss of balance or coordination, changes in hearing or vision, and mood or behavioural changes.

Brain tumours have the potential to be fatal, significantly impacting both patients and their loved ones and changing their lives forever. Due to their invasive nature, they have a high mortality rate. However, detecting them ear can be promising as it can improve the chances of survival. Early diagnosis and detection of brain tumours are crucial for effective treatment and management.

Various imaging techniques such as computed tomography (CT) scans, magnetic resonance imaging (MRI), and positron emission tomography (PET) scans can be utilized to diagnose brain tumours. In addition, a biopsy procedure that involves removing a small tissue sample and examining it under a microscope may be necessary to confirm the diagnosis.

The detection of brain tumours can be automated by combining the YOLOv4 (You Only Look Once version 4) object detection algorithm and a Convolutional Neural Network (CNN) for classification.

2.LITERATURE SURVEY

Mohammad Tariqul Islam et al. [1] proposed a portable electromagnetic (EM) imaging system that utilizes a YOLOv3 deep neural network model. The researchers collected fifty sample images from various head regions and augmented them to create a dataset of 1000 images. The dataset included fifty samples with either single or double tumours. The researchers used 80% of the images for training, 10% for validation, and the remaining 10% for testing the network's performance.

Sethuram Rao.Grish et al. [2], conventional medical imaging methods like Magnetic Resonance Imaging (MRI) may not be able to detect tumours that are smaller than 3mm in size. The authors conducted a review of different Brain Tumour Detection Approaches, including those that use Artificial Neural Network (ANN) and Support Vector Machine (SVM). Our method, which combines YOLO with these techniques, can effectively identify tumours when integrated into the latest technology.

Following a thorough analysis, two major issues are discovered: Image Restoration and Image Enhancement, and proposed a model implemented in Python and TensorFlow to predict brain tumours' quantitative characteristics accurately. Sunil Kumar et al. [3] proposed a CNN-based brain tumour detection approach, which achieved high accuracy and low error rate.

The efficiency of machine learning algorithms in the diagnosis and treatment of brain tumours is explored in several papers, including the work by Mohammad Omid Khairandish et al [4]. The studies evaluate the algorithm type, dataset, proposed model, and performance, with accuracy levels ranging from 79% to 97.7%. The most utilized algorithms are CNN, KNN, C-means, and RF, in descending order of frequency.

Parveen et al. [5] developed a novel hybrid technique for brain tumour classification using support vector machine (SVM) and fuzzy c-means. The proposed algorithm improves the quality of MRI images by using contrast enhancement and mid-range stretch techniques, which is like the approach used in [2]. Magnetic resonance imaging (MRI) remains the most crucial method for detecting brain tumours.

Mohammad Shahjahan Majib et al. [6] introduces a deep learning framework called "VGG-SCNet" that is based on the VGG Net architecture for detecting brain tumours in MRI images. The main focus is on accurately segmenting and diagnosing tumour areas in MR images. The authors also evaluated 16 different transfer learning models to determine the optimal model for neural network-based classification of brain tumours.

M.O. Khairandish et al. [7] utilized a hybrid technique that combines CNN and SVM models to identify and classify brain tumours in MRI images. Their approach differs from the methods proposed in other studies [2-6]. The results suggest that their approach significantly improves the accuracy of tumour detection.

Simon Podnar et al. [8] found that routine blood tests may contain more valuable information than what clinicians usually perceive. A machine learning predictive model that diagnoses brain tumours using routine blood tests. The authors tested the model using a retrospective analysis of 68 consecutive brain tumour patients and 215 control patients who visited the neurological emergency service. The adapted tumour model demonstrated a sensitivity of 96% and specificity of 74% in the validation group.

3.PROBLEM STATEMENT

The health-care industry is distinct from other sectors. Because it is a high priority sector, people want the best care and service available, regardless of cost. Because of its success in other real-world applications, deep learning is now providing innovative solutions with high accuracy for medical imaging and is a significant technique for upcoming applications in the health sector.

Detecting an automated brain tumour in an MRI is difficult due to its complex position and size variations. Brain tumours are identified and classified using YOLO (You Only Look Once), a cutting-edge object recognition framework.

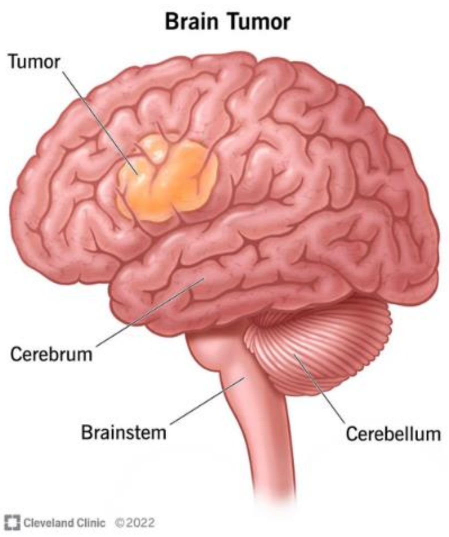

Fig.3.1. Brain Tumour

4.SYSTEM ANALYSIS

4.1 EXISITING SYSTEM

Machine learning (ML) algorithms have been developed in recent years to aid in the detection and diagnosis of brain tumours. These methods create models that can detect the presence and characteristics of brain tumours using data from imaging studies such as CT scans and MRI. One ML method for brain tumour detection is to use a convolutional neural network (CNN) to analyse brain images and classify them as normal or containing a tumour. Another example is analysing electronic medical records using natural language processing (NLP) to identify patients with brain tumours based on their symptoms and diagnostic test results. These machine learning-based methods have the potential to be faster, more accurate, and less invasive than traditional methods for detecting brain tumours.

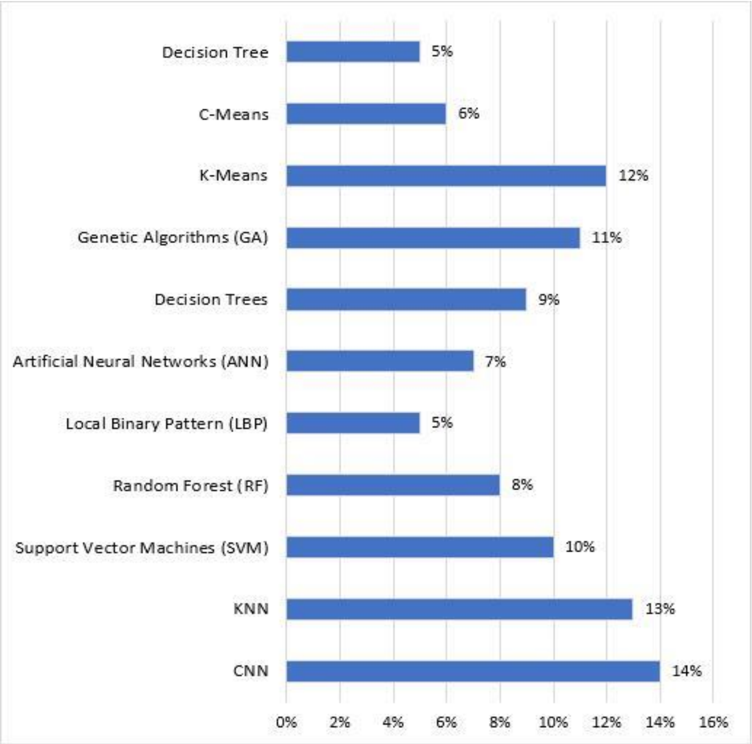

Fig.4.1. Method wise trends since 2012–2019

4.2 PROPOSED SYSTEM

The YOLOv4 (You Only Look Once version 4) object detection algorithm is combined with a Convolutional Neural Network (CNN) for classification. YOLOv4 is a cutting-edge object detection algorithm capable of detecting objects in images and video in real time. It works by dividing an image into grid cells and predicting the presence and type of objects in each.

A CNN is a type of neural network designed specifically for image classification tasks. It can learn features from images and use them to classify images into various categories. In the case of brain tumour detection, the CNN would be trained to classify images as having or not having a brain tumour.

It is possible to detect brain tumours accurately and efficiently by using YOLOv4 for object detection to identify potential tumours in an image and a CNN for classification to confirm the presence of a tumour.

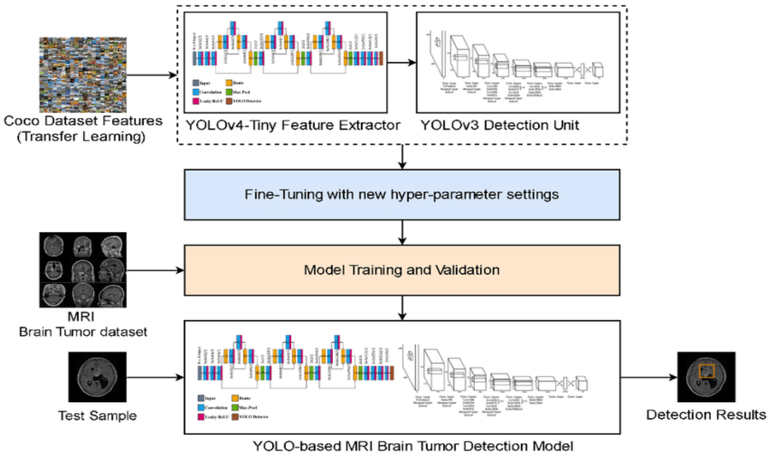

Fig.4.2.1. YOLO V4 Model

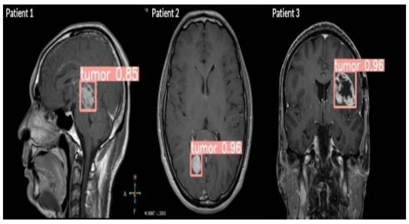

Fig.4.2.2 MRI images of Brain Tumour

4.3 SYSTEM REQUIREMENTS

4.3.1 SOFTWARE REQUIREMENTS:

- Python libraries and Packages: numpy, pandas, Matplot, sklearn, darknet, keras, TensorFlow, pytorch, OpenCV
- Software: Mathlab

4.3.2 HARDWARE REQUIREMENTS:

- Operating system: Windows
- Processor: Minimum intel i5
- Ram: Minimum 4 Gb
- Hard disk: Minimum 256 Gb

4.3.3 FUNCTIONAL REQUIREMENTS:

- Data collection
- Data pre-processing
- Training and testing
- Detecting
- Classifying

4.3.4 NON-FUNCTIONAL REQUIREMENTS:

- Availability requirement
- Scalability requirement
- Reliability requirement
- Data integrity requirement
- Manageability requirement

5.METHODOLOGY

- **Data Acquisition**: There is a manual dataset which consists of Images of different MRI taken from different datasets, and of different sizes. The data in this dataset is less efficient for performing the detection and classification of brain tumours. So, it requires multiple pre-processing steps to get all images to a efficient format for training a model.
- **Data Pre-processing:** The steps in pre-processing pipeline are.
  - **Training Model using YOLO Model:** To detect tumours in MRI images using YOLO (You Only Look Once), first need to train the YOLO algorithm on a dataset of MRI images that contain tumours and do not contain tumours. This would involve labelling the images with bounding boxes around the tumours and assigning class labels (e.g., "tumour" or "no tumour") to each image.
  - **Pre-process the MRI image:** YOLO uses convolutional neural networks (CNNs) to extract features from the input image. The CNN is trained to recognize features that are indicative of the presence of an object, such as edges, corners, and patterns. This may involve resizing or scaling the image to the appropriate size and format for the YOLO algorithm.

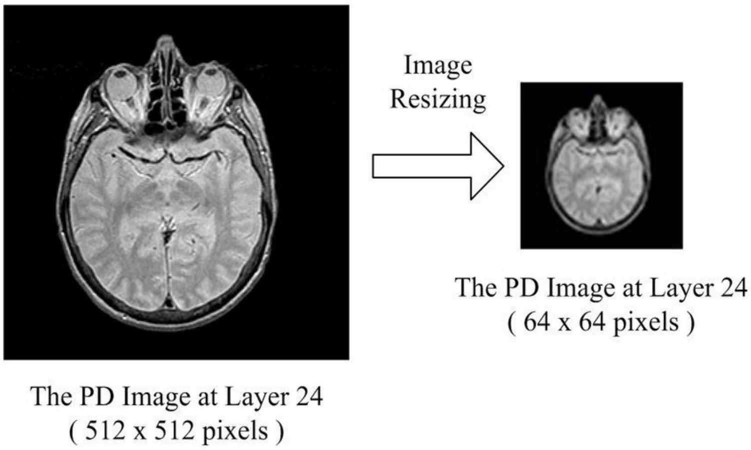

Fig.5.1. Image Resizing

  - **Object detection:** YOLO uses a combination of bounding boxes and class probabilities to detect objects in the image. A bounding box is a rectangle that surrounds an object and is defined by its top-left and bottom-right coordinates. The class probability is the probability that the object belongs to a particular class (e.g., brain tumour).

- **Confidence Score:**

Confidence Score = Pr(object) * IoU

Where Pr(object) is conditional class probability and IoU is the Intersection-over-Union (IoU) overlap between the proposal and the ground truth box.

- **Softmax Function:**

softmax(x) = exp(x) / sum(exp(x))

Where x is a vector of scores and exp(x) is the element-wise exponentiation of x.

- **Non-maximal Suppression:** YOLO uses non-maximal suppression to eliminate redundant or overlapping bounding boxes and improve the accuracy of the object detection. This involves comparing the class probabilities and overlap of the bounding boxes and keeping only the ones with the highest-class probability and the least overlap with other boxes. **Non-maximal suppression** is also called **Masking.**

Non-Maximum Suppression (NMS) algorithm

Input

Bounding box proposals: P = {p_1, p_2, ..., p_n} where p_i = (x_i, y_i, w_i, h_i) represents the center coordinates (x, y) and width and height (w, h) of the i-th bounding box proposal.

Confidence scores: C = {c_1, c_2, ..., c_n} where c_i represents the confidence score of the i-th bounding box proposal

IoU threshold: t (usually set to 0.5)

Output

Final set of bounding box proposals: Q= {q_1, q_2, ..., q_m} where m <= n and each q_i is a selected proposal

Algorithm

Sort the bounding box proposals in decreasing order of their confidence scores.

Initialize an empty list Q to store the selected proposals.

While there are still proposals in P:

a. Select the proposal with the highest confidence score and add it to Q.

b. Remove all proposals from P that have an IoU overlap with the selected proposal greater than or equal to the threshold t.

Return the final set of selected proposals Q.

The formula for computing the IoU overlap between two bounding box proposals p_i and p_j is:

IoU(p_i, p_j) = Area of Intersection(p_i, p_j) / Area of Union(p_i, p_j)

where the Area of Intersection(p_i, p_j) is the area of the region where the two proposals overlap and the Area of Union(p_i, p_j) is the sum of the areas of the two proposals minus their intersection.

- Masking acts as a filter and as instance segmentation (real-time segmentation). By Masking, identification of tumour in MRI image is easy.

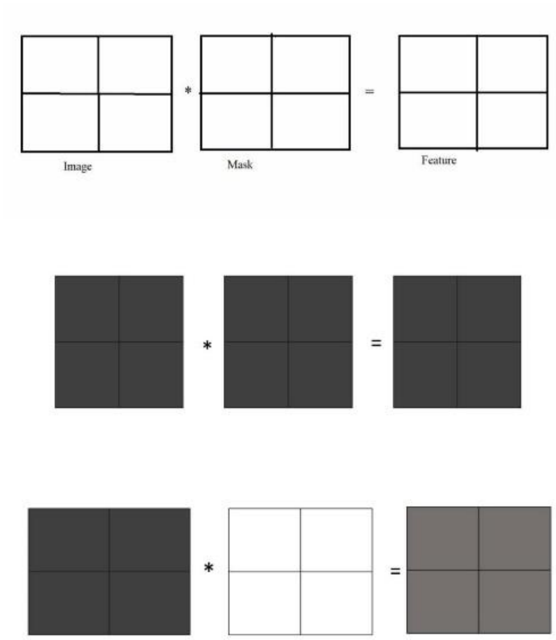

Fig.5.2. MASKING

- **Anchor boxes**: They are a set of predefined boxes of different sizes and aspect ratios that are used to predict the coordinates of the bounding box proposals.

The center coordinates (x, y) of the bounding box proposal relative to the top-left corner of the grid cell are predicted as offsets from the grid cell coordinates.

$$x = \sigma(t_x) + c_x$$

$$y = \sigma(t_y) + c_y$$

The width and height of the bounding box proposal are predicted as offsets from the anchor box sizes.

$$w = s_e^{\sigma(t_w)}$$

$$h = s_e^{\sigma(t_h)}$$

The formula for the sigmoid function used to map the predicted offsets to values between 0 and 1 is:

$$\sigma(x) = 1 / (1 + e^{-x})$$

- **Classify the detected objects:** You can use the class probabilities output by YOLO to classify the detected objects as either tumours are present or not.

- **Classifying using CNN Model:** A CNN is a type of neural network specifically designed for image classification tasks. It can learn features from images and use them to classify the images into different categories. In the case of brain tumour detection, the CNN would be trained to classify images as either containing a brain tumour or not.

CNN-based Object Detectors are most used in recommendation systems. YOLO (You Only Look Once) models are used for high-performance object detection. YOLOV4 divides an image into grids, each of which detects objects within itself. Based on the data streams, they can be used for real-time object detection. They require very little computational power. (As illustrated in fig.5.3)

History of YOLO

1. Yolov1(Jun 8th, 2015): You Only Look Once: Unified, Real-Time Object Detection
2. Yolov2 (Dec 25th, 2016): YOLO9000:Better, Faster, Stronger
3. Yolov3 (Apr 8th, 2018): YOLOv3: An Incremental Improvement
4. Yolov4 (Apr 23rd, 2020): YOLOv4: Optimal Speed and Accuracy of Object Detection

- **YOLO v4 Model Architecture**
  - Model Backbone
  - Model Neck
  - Model Head

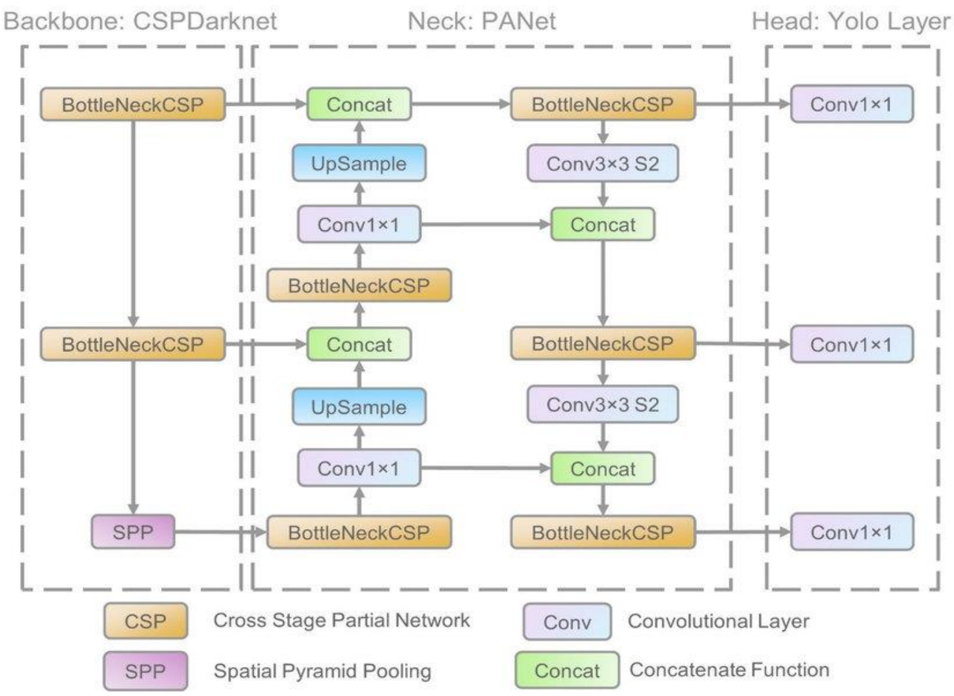

Fig.5.3. YOLO V4 Architecture

WORKING OF YOLO ALGORITHM

YOLO algorithm works using the following three techniques:

- ❑ Residual blocks
- ❑ Bounding box regression
- ❑ Intersection Over Union (IOU)

Residual blocks:

The image is divided into grids. Each grid has a dimension of S x S. The image below shows how an input image is divided into grids.(as shown in fig.5.4)

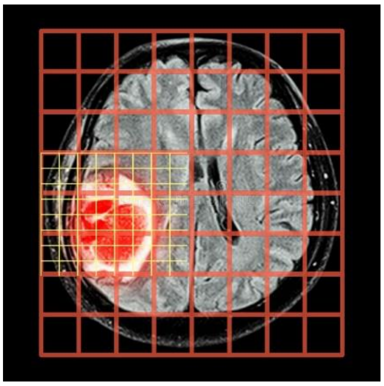

Fig.5.4. Residual Blocks

Bounding box regression:

A bounding box is an outline that draws attention to an object in an image. (As illustrated in Fig5.5)

Every bounding box in the image consists of the following attributes:

- ❑ Width (bw)
- ❑ Height (bh)
- ❑ Class
- ❑ Bounding box center (bx,by)

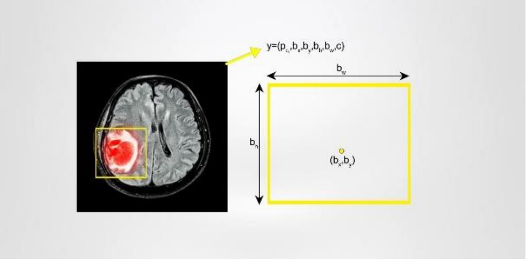

Fig.5.5. Bounding Box Regression

Intersection over union (IOU):

Intersection over union (IOU) is an object detection phenomenon that describes how boxes overlap. YOLO employs IOU to create an output box that perfectly surrounds the objects. Each grid cell oversees predicting the bounding boxes as well as their confidence scores. If the predicted and actual bounding boxes are the same, the IOU is equal to one. This mechanism removes bounding boxes that are not the same size as the actual box. The image below shows a simple example of how IOU works.

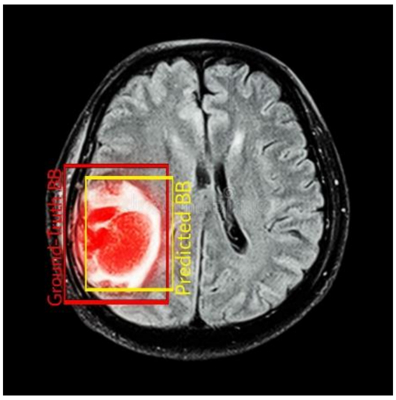

Fig.5.6. Intersection Over Union

There are two bounding boxes in the image above, one in red and one in yellow. The yellow box represents the predicted box, while the red box represents the actual box. YOLO checks to see if the two bounding boxes are equal. (As illustrated in Fig.5.6)

Combination of three techniques:

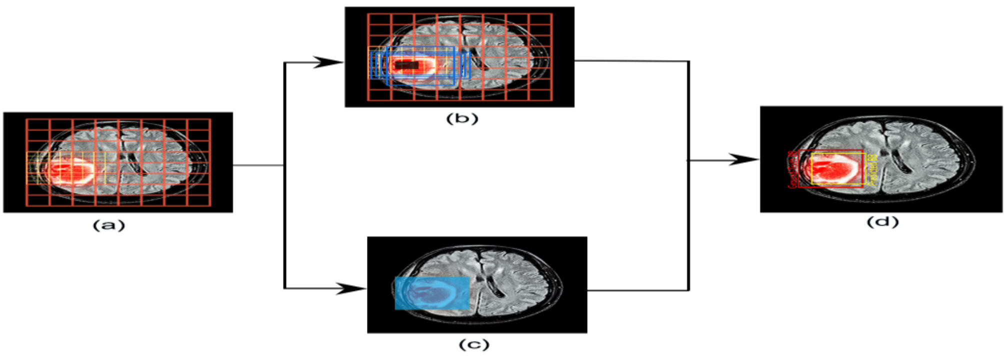

Fig.5.7. Combination of three techniques

6.ARCHITECTURE

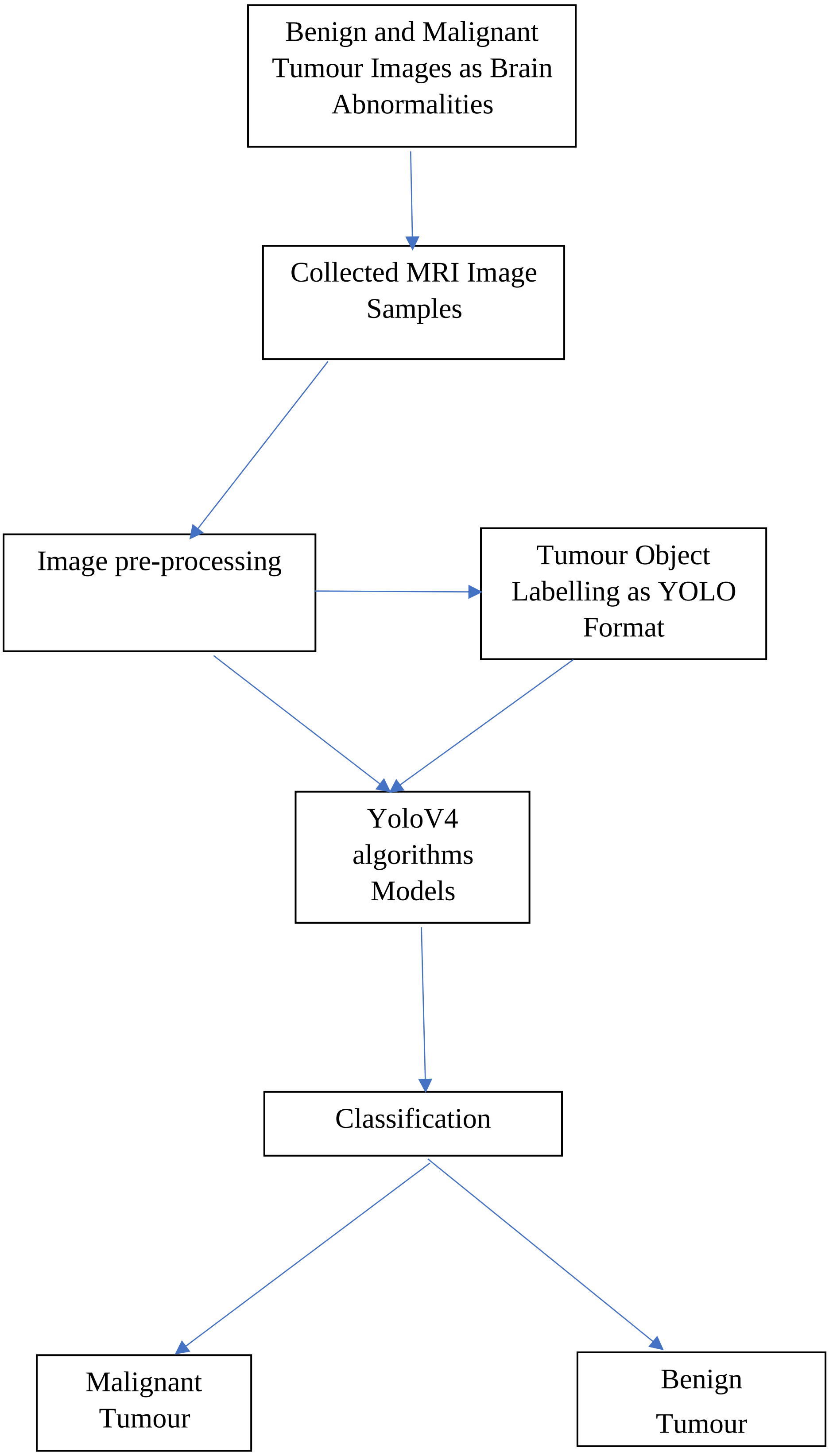

Fig.6.SYSTEM ARCHITECTURE

7.SYSTEM DESIGN

The chapter describes in detail the entire design of the model and system framework. System design is defined as "the process of designing a system's architecture, components, and interfaces to meet end-user requirements." Unified Modeling is used in this work to clearly describe the model.

UML Diagrams

UML is an abbreviation for Unified Modeling Language. In the field of object-oriented software engineering, UML is a standardised general-purpose modelling language. The Object Management Group manages and created the standard.

The Primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modelling Language that allows them to create and exchange meaningful models.
2. Provide mechanisms for extensibility and specialisation to extend the core concepts.
3. Be independent of programming languages and the development process.
4. Establish a formal foundation for understanding the modelling language.
5. Promote the expansion of the market for OO tools.
6. Encourage the use of higher-level development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices.

The main UML diagrams can be categorized into the structure and behavioural diagrams as illustrated in the bellow figure.

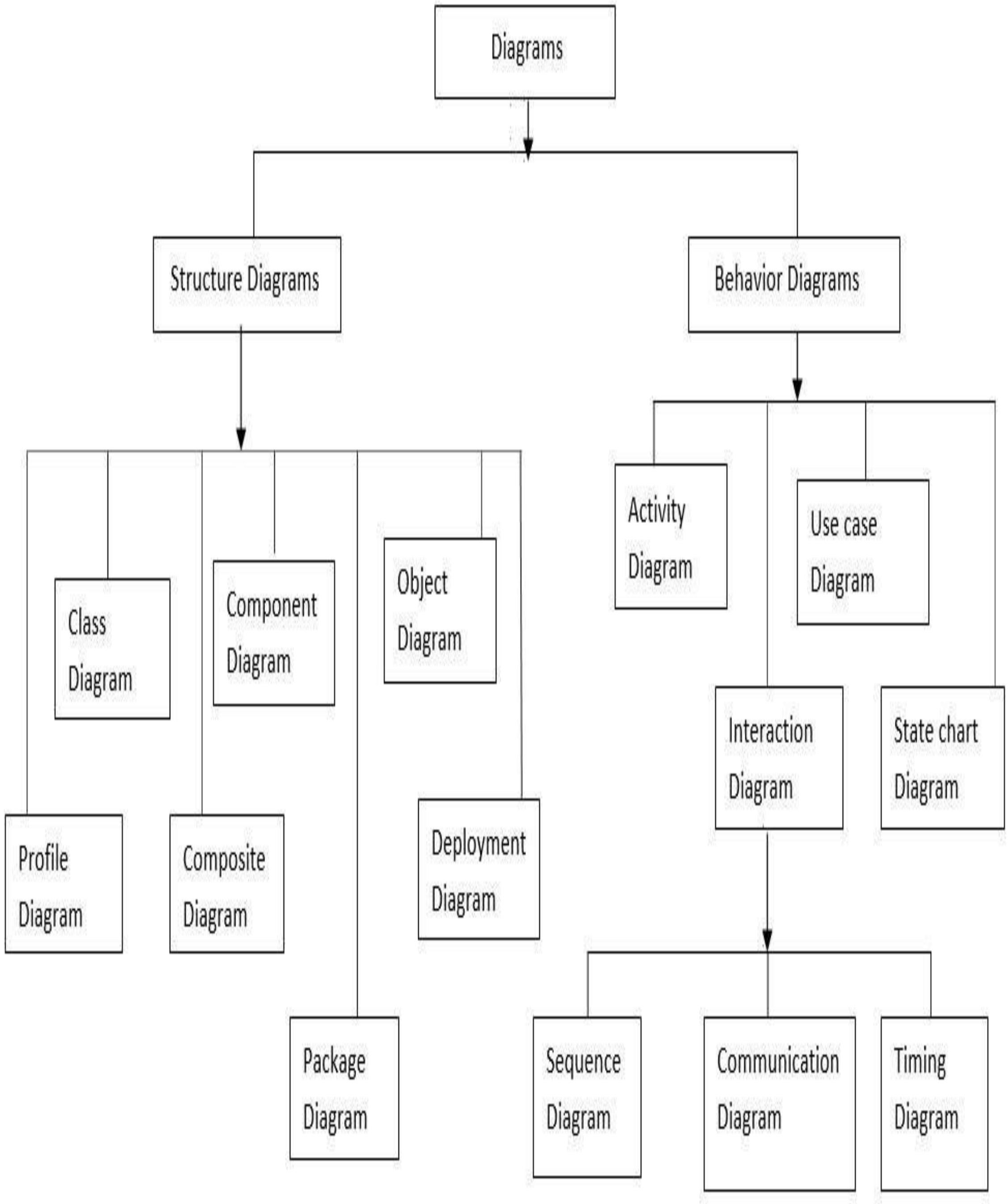

Fig.7.1.UML DIAGRAMS CLASSIFICATION

USECASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram is a type of behavioural diagram defined by and created from a use-case analysis. Its goal is to provide a graphical overview of a system's functionality in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. A use case diagram's main purpose is to show which system functions are performed for which actor. The roles of the system's actors can be depicted.

The use case diagram for our framework is:

Fig.7.2. USECASE DIAGRAM

SEQUENCE DIAGRAM

In Unified Modeling Language (UML), a sequence diagram is a type of interaction diagram that shows how processes interact with one another and in what order. It is a Message Sequence Chart construct. Event diagrams, event scenarios, and timing diagrams are other names for sequence diagrams.

The model's sequence diagram is as follows:

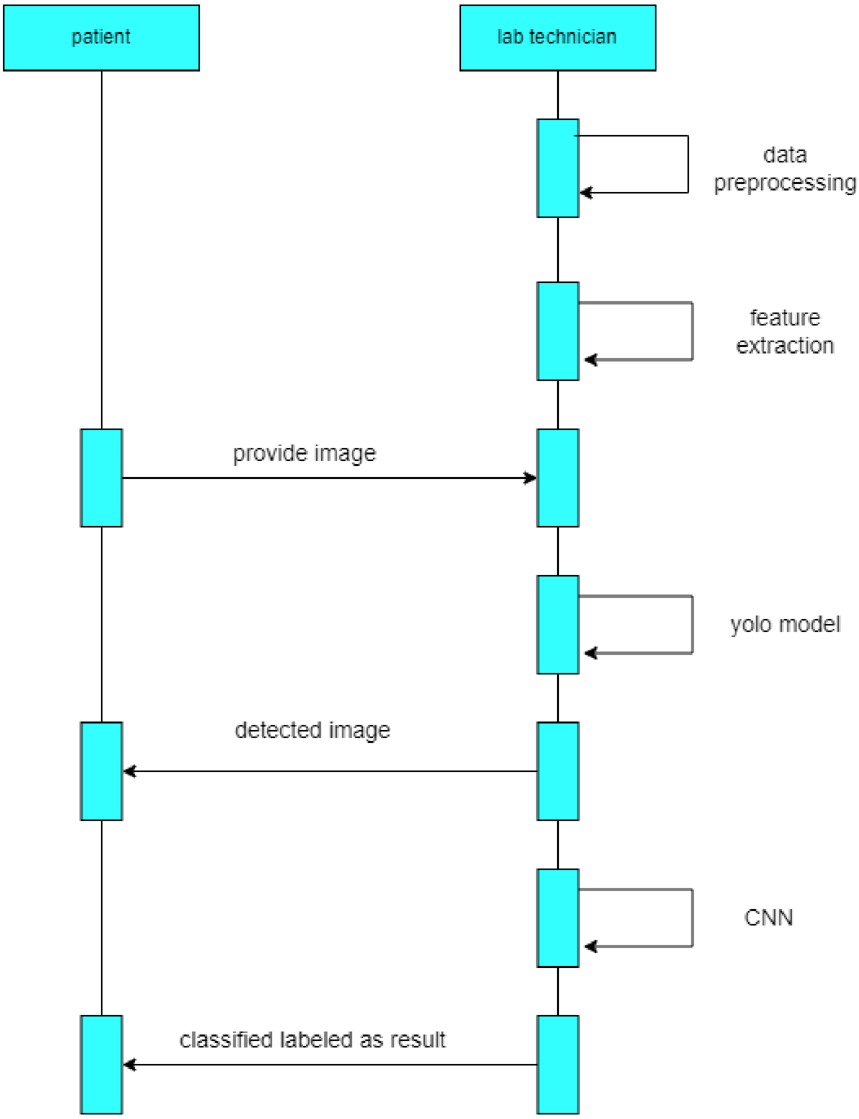

Fig.7.3. SEQUENCE DIAGRAM

STATECHART DIAGRAM

The state machine UML diagrams, also known as Statechart diagrams, are used to describe the various states of a system component. The diagram is called a state machine because it is essentially a machine that describes the various states of an object and how they change in response to internal and external events.

The model's statechart diagram is as follows:

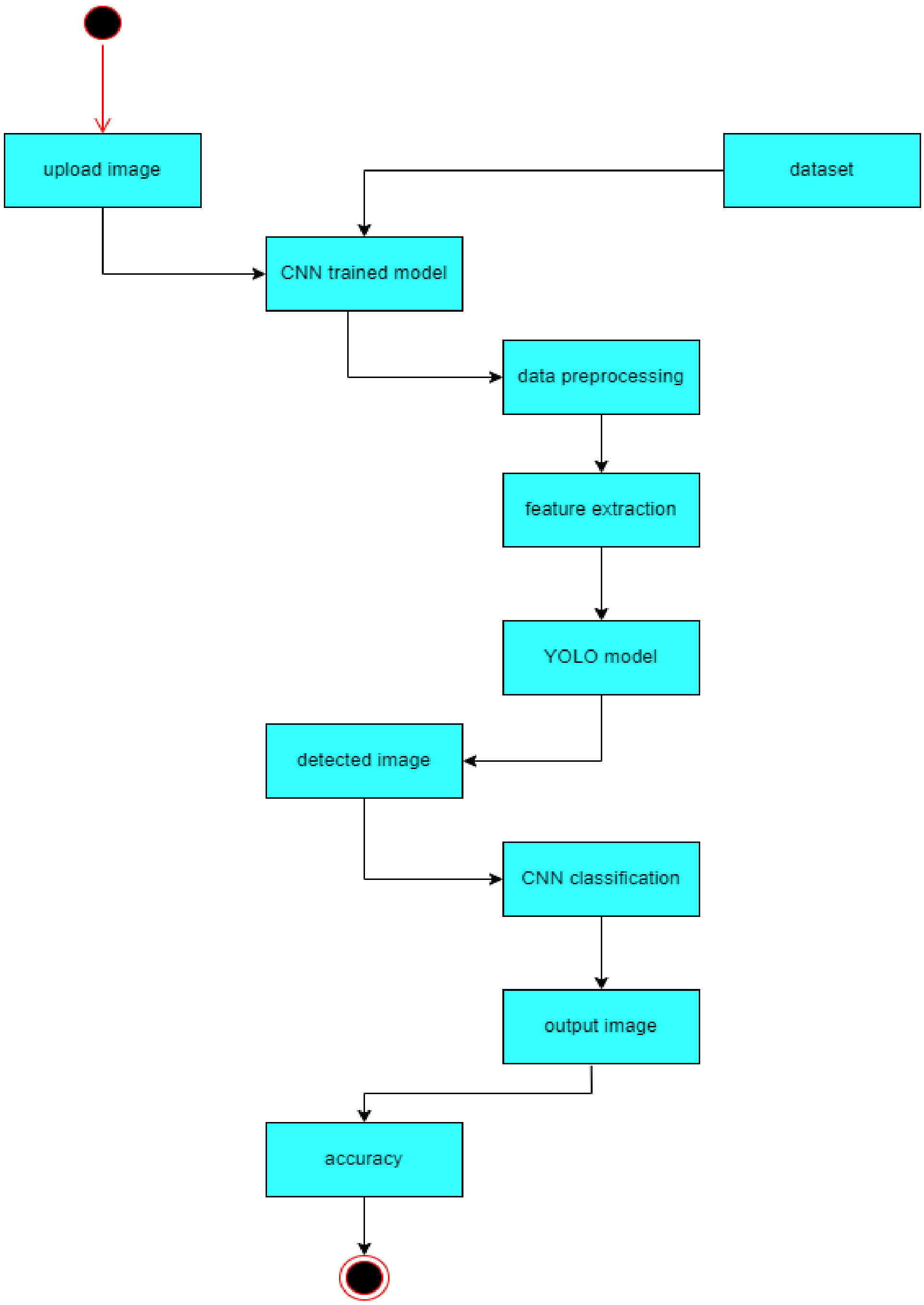

Fig.7.4. STATECHART DIAGRAM

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of sequential activities and actions that allow for selection, iteration, and concurrency. Activity diagrams in the Unified Modeling Language can be used to describe the business and operational step-by-step workflows of system components. An activity diagram depicts the overall control flow.

The model's activity diagram is as follows:

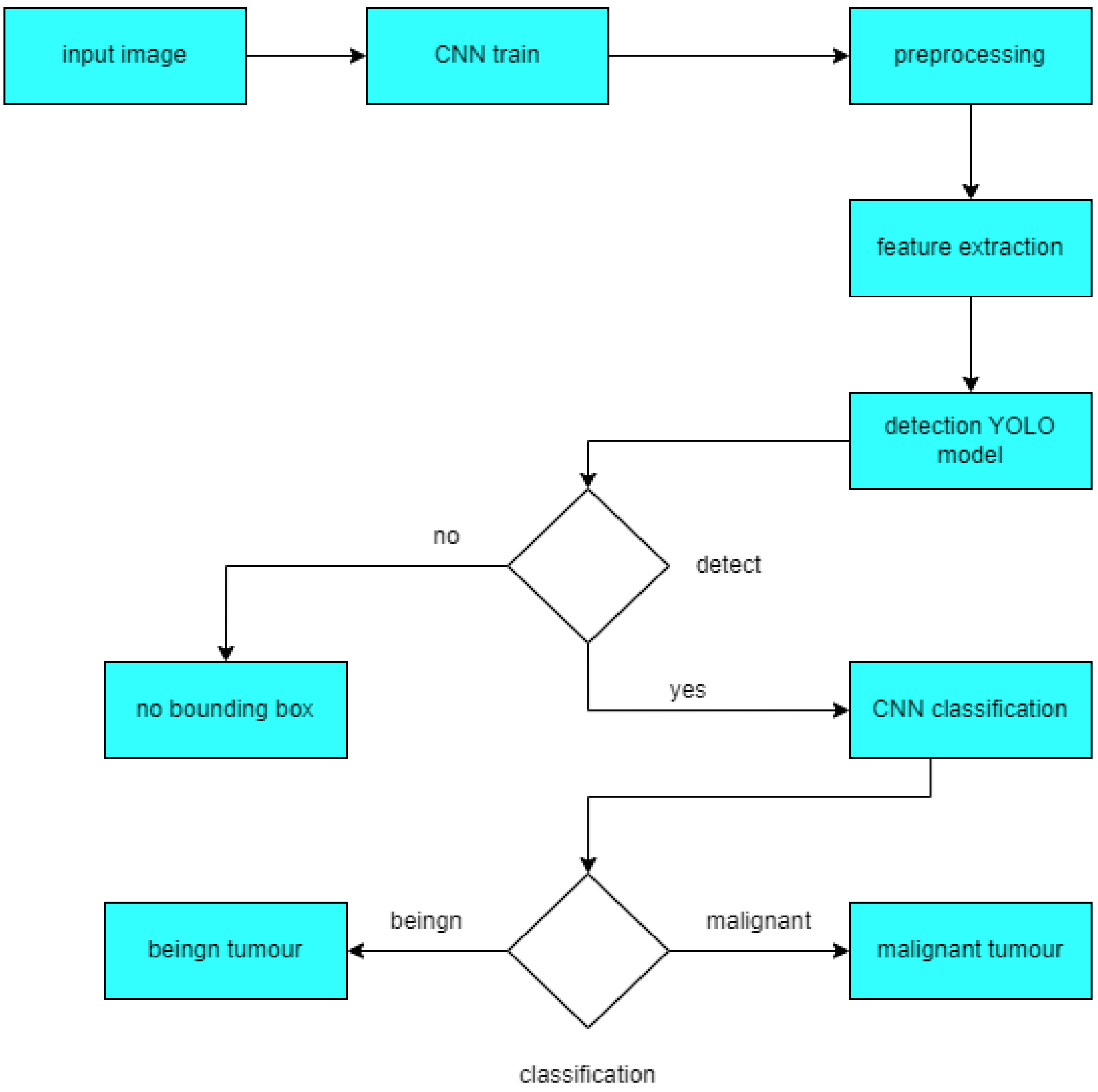

Fig.7.5. ACTIVITY DIAGRAM

8.IMPLEMENTATION

The execution is done by using a software called MATLAB. Below are the steps for implementing a machine learning model along with feature extraction.

DATA COLLECTION

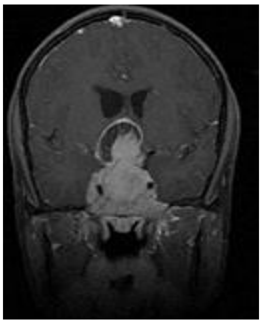

Fig.8.1.MRI IMAGE

Figure 9.1 shows the example of an image in the dataset which is collected from various sources an open dataset from Kaggle website. The dataset is fed into a model for training in order to improve the model's performance. The dataset includes large set of images, with every set consisting of different types of brain tumours

DATA PREPROCESSING

THRESHOLDING

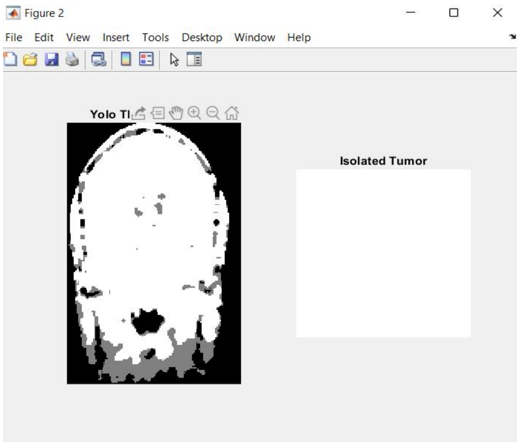

Fig.8.2. THRESHOLDING

Thresholding is a common image processing and computer vision technique for converting a grayscale or colour image to a binary image. For a binary image, the process entails selecting a threshold value and then setting all pixel values below that threshold to zero (black) and all values above the threshold to 255 (white).

GREY SCALE CONVERSION

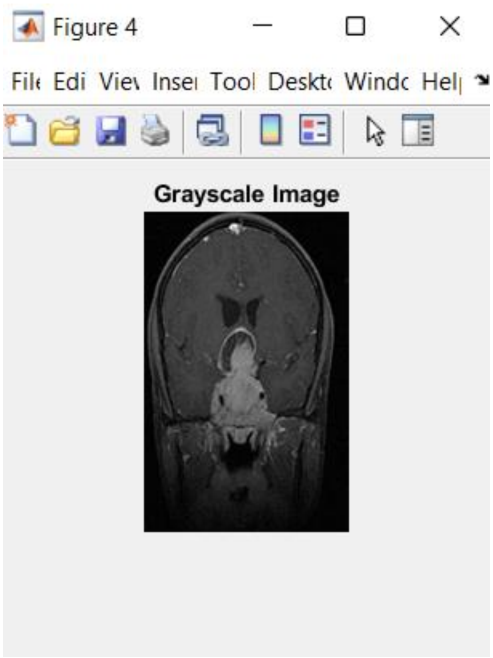

Fig.8.3. GRAYSCALE CONVERSION

Figure 9.2 shows the screenshot of grey scale conversion of original input image for better understanding the features of an image in the dataset.

HIGH PASS FILTER

Fig.8.4. HIGH PASS FILTER

Most sharpening methods are based on a high pass filter. When contrast between adjacent areas with little variation in brightness or darkness is increased, an image is sharpened. A high pass filter tends to retain high frequency information while reducing low frequency information in an image.

TRAINING PROGRESS

When training a machine learning model, the parameters of the model are updated to minimise the loss function on the training data. Typically, the training process entails feeding the model batches of training examples and using an optimizer algorithm to update the model's parameters in the direction of the gradient of the loss with respect to the parameters. This process is typically repeated multiple times, with each epoch iterating over the entire training dataset.

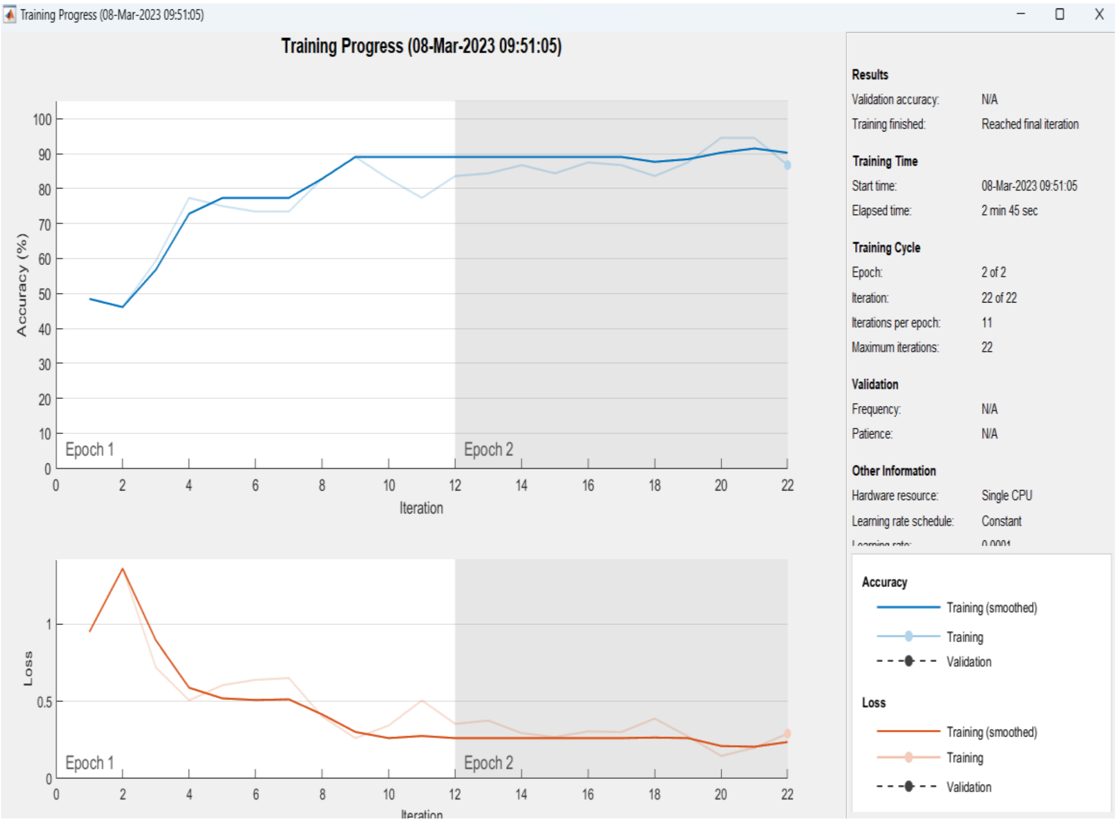

Fig.8.5. TRAINING PROGRESS B/W ACCURACY AND LOSS

YOLO DETECTION

YOLO (You Only Look Once) is a real-time object detection system that uses a single neural network to detect objects in images and video. Unlike traditional object detection methods, which perform region proposals and classification separately, YOLO performs both tasks in a single step, resulting in faster and more accurate detection.

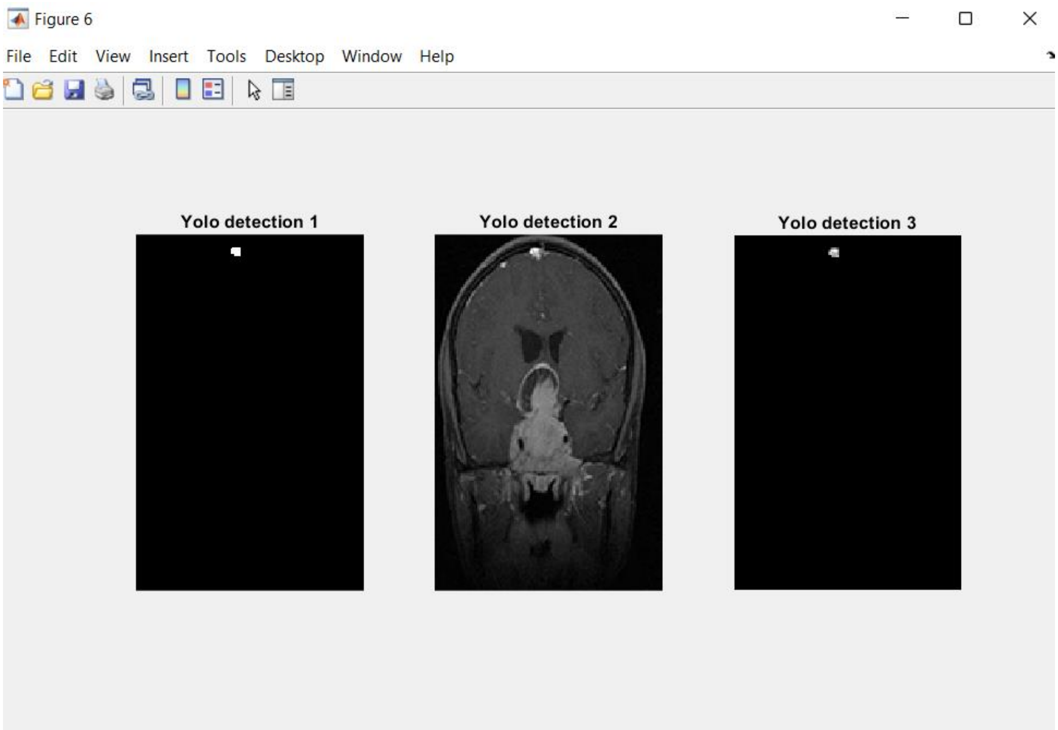

Fig.8.6. YOLO DETECTORS

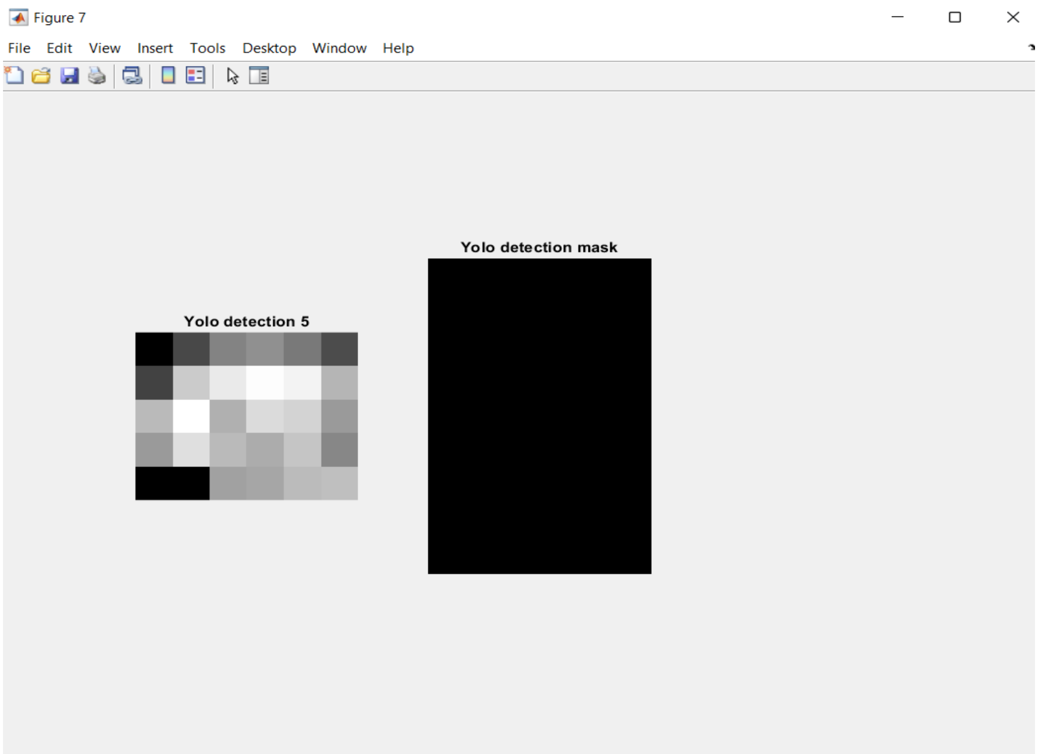

Fig. 8.7. MASKING

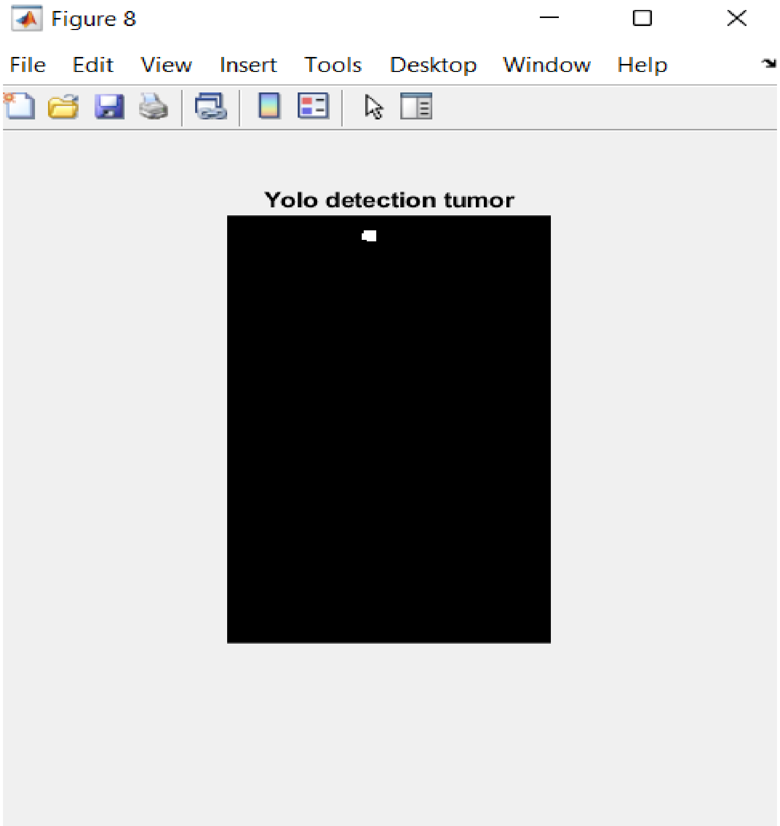

Fig.8.8. YOLO DETECTED TUMOUR AFTER MASKING

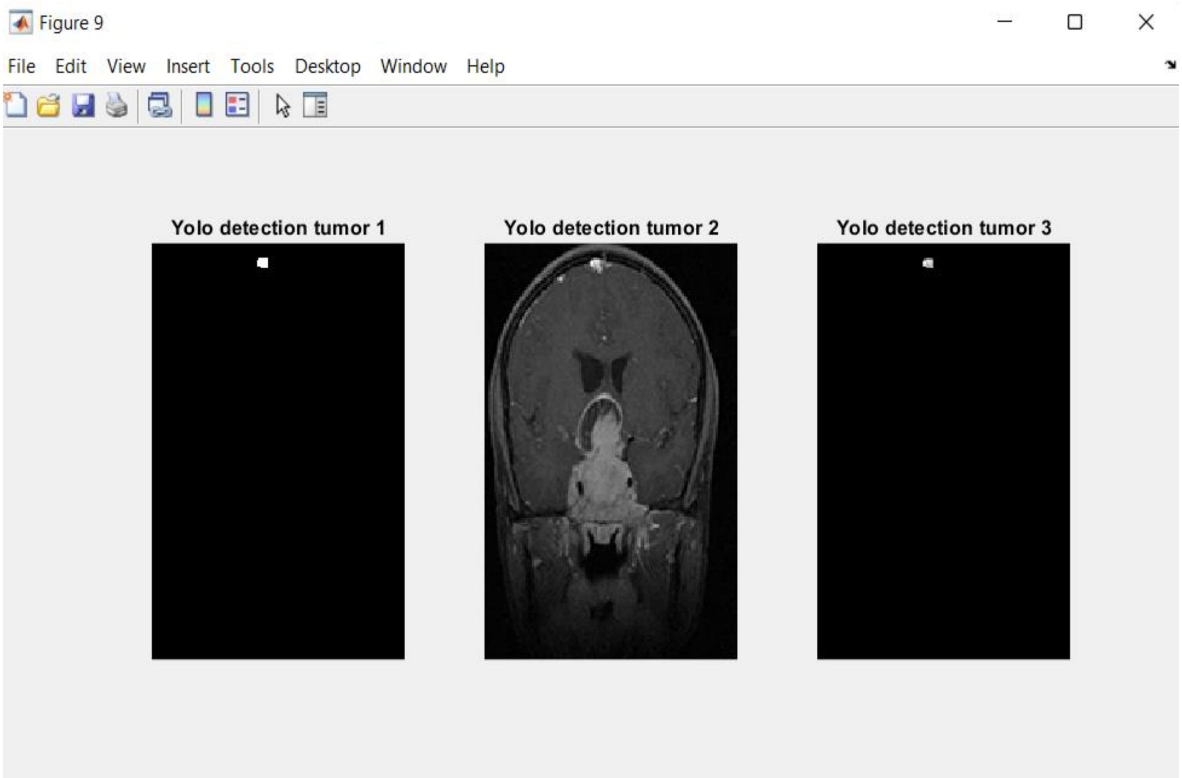

Fig.8.9. YOLO DETECTORS AFTER MASKING

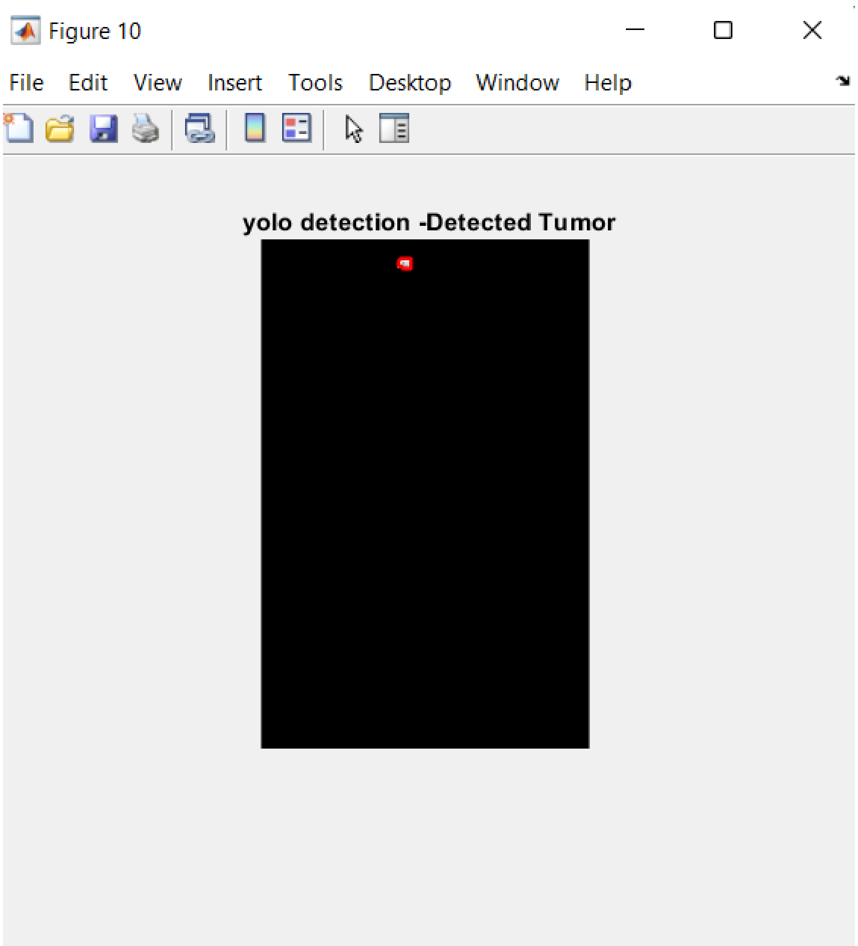

Fig.8.10. DETECTED TUMOUR WITH BOUNDARY

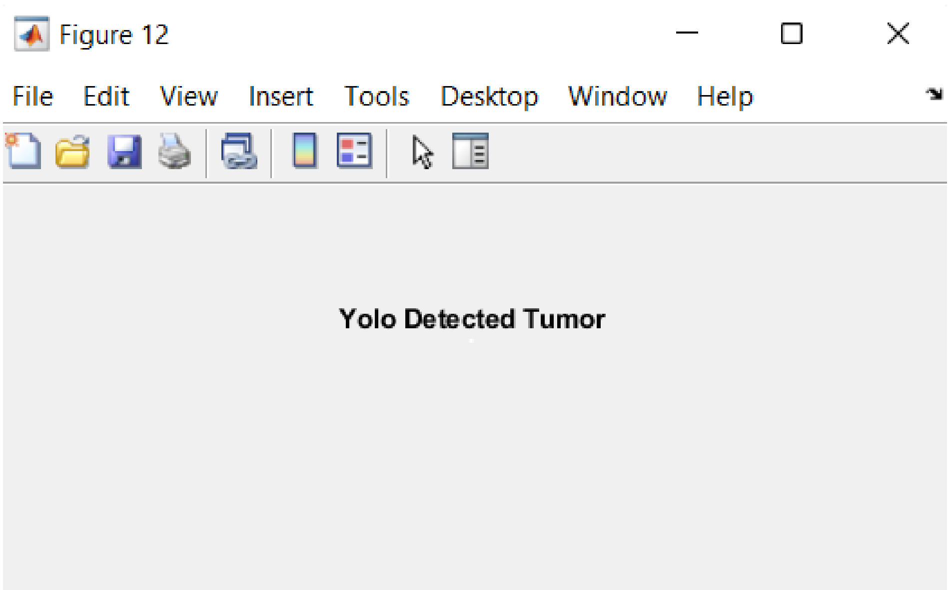

Fig.8.11. TUMOUR DETECTED USING YOLO

LOCALIZATION

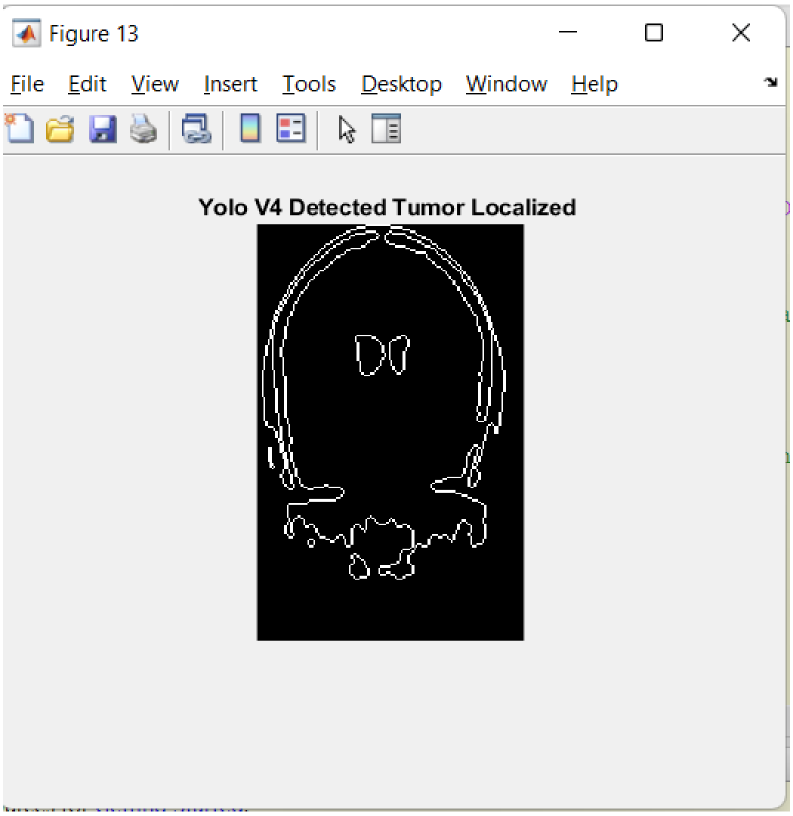

Fig.8.12. LOCALIZATION

Localization refers to the task of identifying the location or spatial extent of an object or region of interest within an image.

CNN CLASSIFICATION

CNN (Convolutional Neural Network) classification is a type of machine learning algorithm used for image classification tasks. It is a deep learning architecture that is particularly effective in identifying features within images and learning to classify them.

The basic idea behind CNN classification is to extract features from an input image using a series of convolutional layers. These convolutional layers apply a series of filters to the image, each of which detects a different pattern or feature. The convolutional layers' outputs are then passed through one or more fully connected layers, which learn to classify the image based on the extracted features.

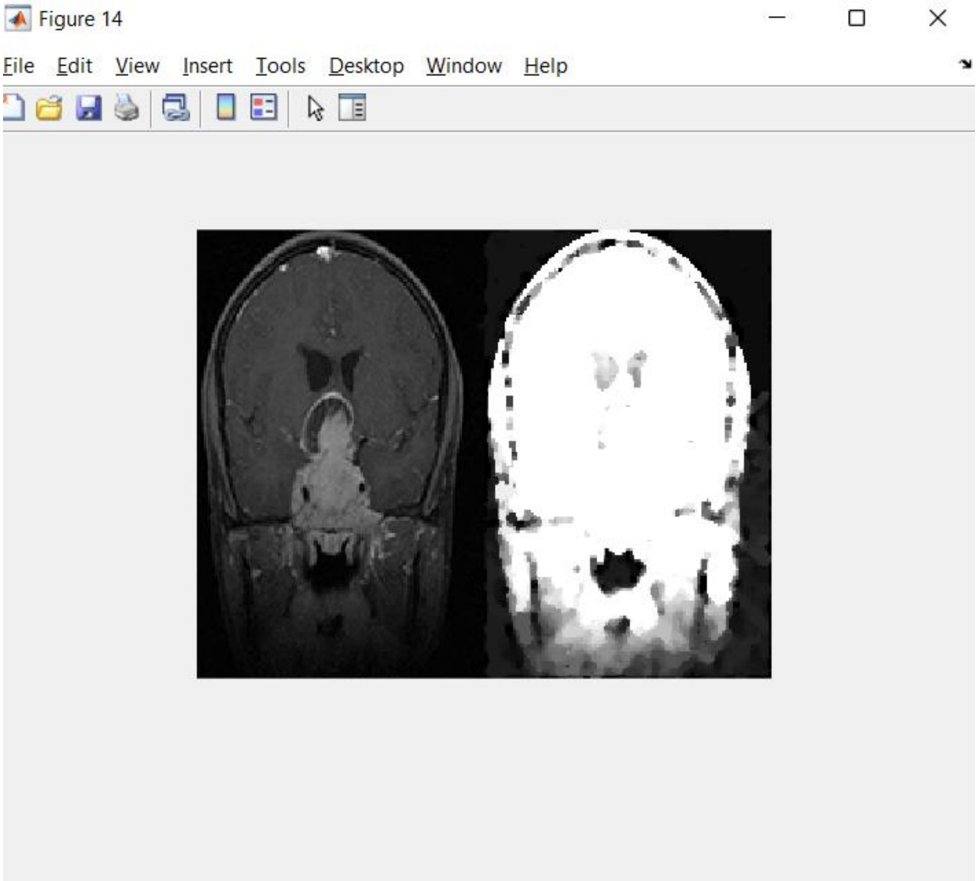

Fig.8.13.CNN CLASSIFICATION

Fig.8.14. BENIGN TUMOUR

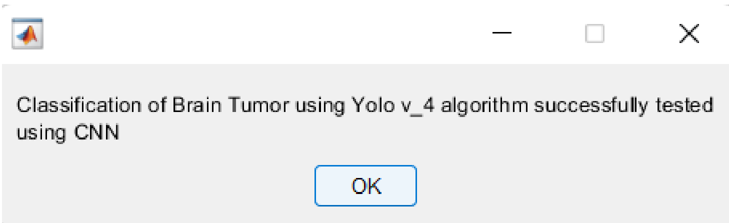

Fig.8.15. CLASSIFIED USING CNN AND DETECED USING YOLO

9.RESULT ANALYSIS

The result analysis can be implemented by using 5 different classification techniques. They are YOLO+CNN, Random Forest, KNN , ANN , Decision Tree. After analysis the performance of these algorithms using evaluation metrics like accuracy. They can be found using confusion matrix and roc curves. The results for the below classification techniques accuracy's are 90-95% for Yolo+CNN , 80% for Random Forest, 55.6% for KNN , 55.6% for ANN , 73.3% for Decision Tree. The performance of metrics and roc curves for algorithms are described below.

Performance of Metrics

Training Progress of YOLO+CNN

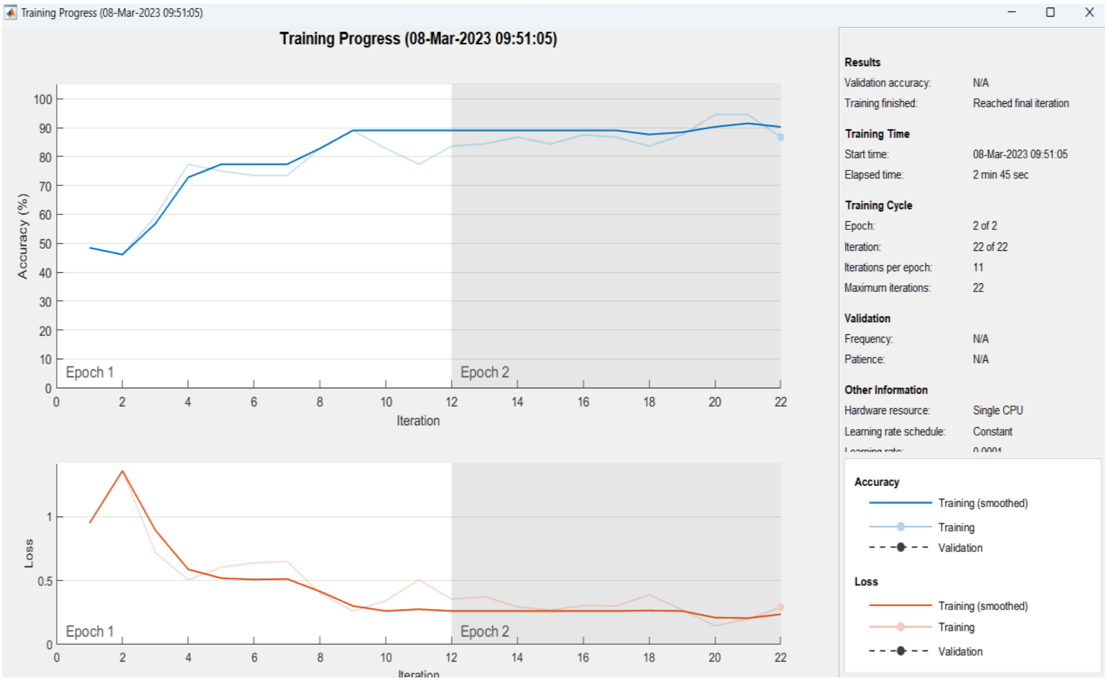

Fig. 9.1 Training Progress of YOLO-CNN

Accuracy for YOLO+CNN

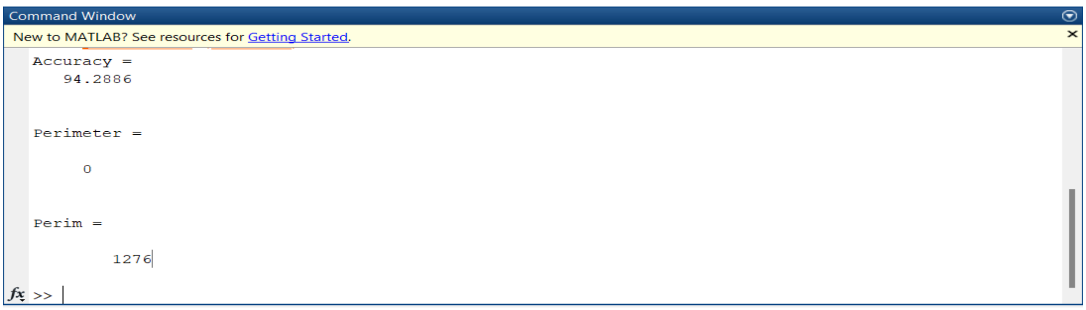

Fig.9.2 Accuracy for YOLO-CNN

ROC Curve for Random Forest

The ROC Curve for Random Forest is depicted in the figure below.

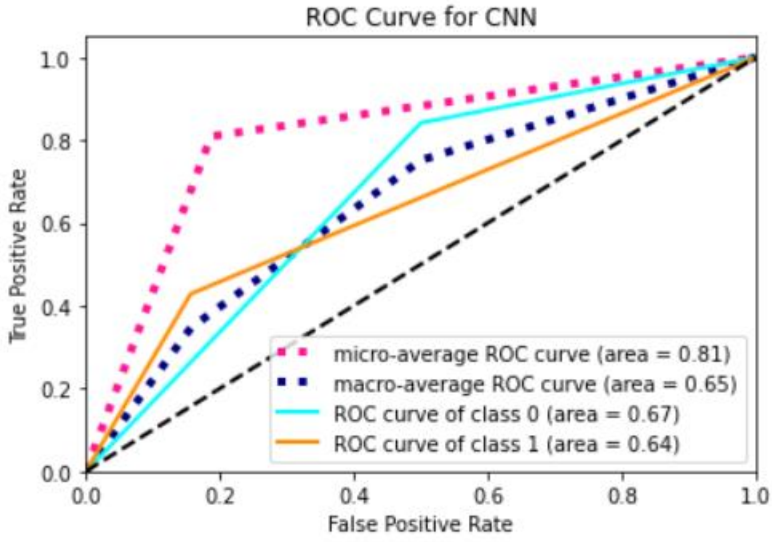

Fig. 9.3 ROC Curve for Random Forest

Confusion Matrix for Random Forest

The Random Forest Confusion Matrix is depicted in the figure below.

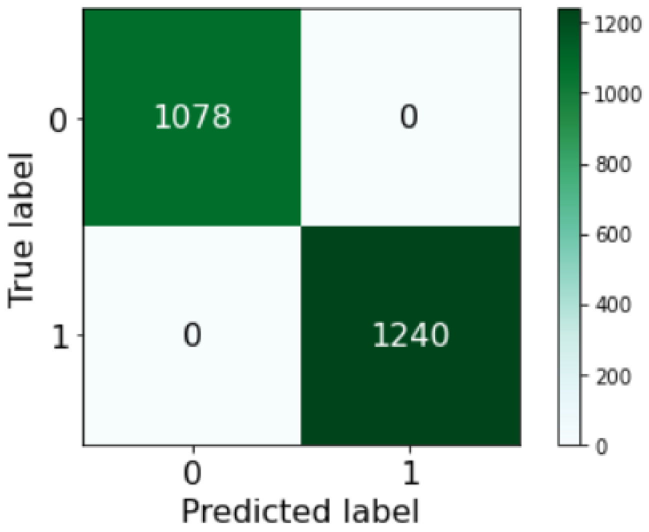

Fig. 9.4 Confusion Matrix for Random Forest

ROC Curve for KNN

The ROC Curve for KNN is depicted in the figure below.

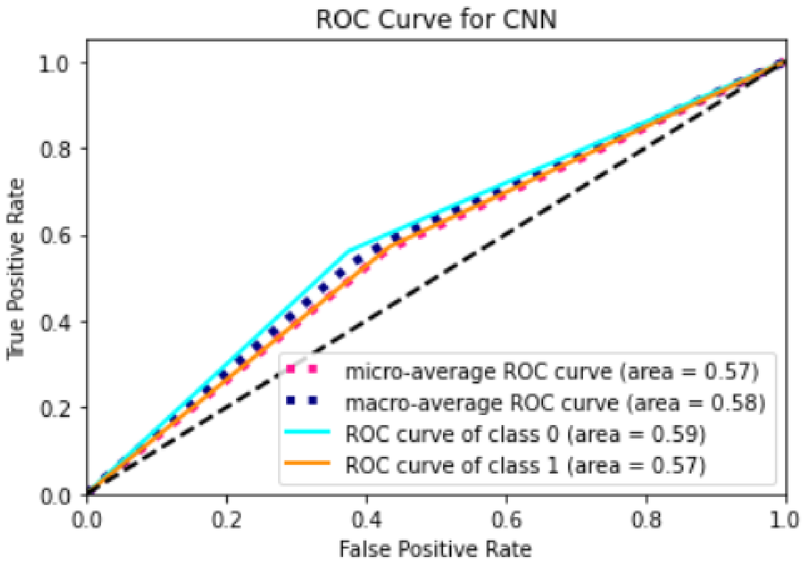

Fig. 9.5 ROC Curve for KNN

Confusion Matrix for KNN

The KNN Confusion Matrix is depicted in the figure below.

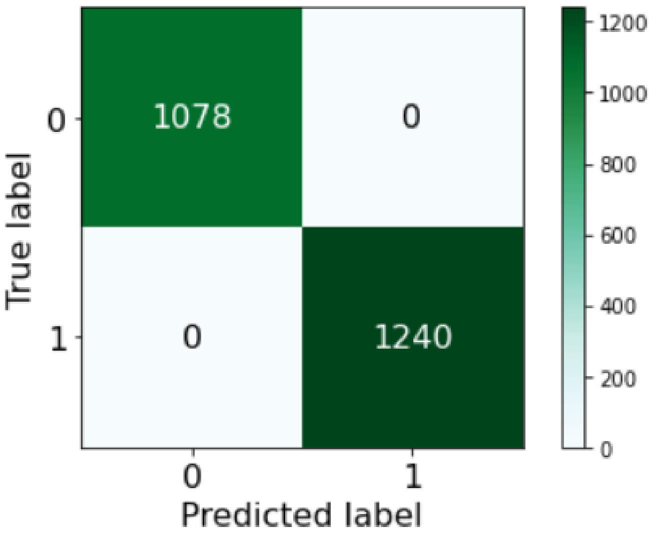

Fig. 9.6 Confusion Matrix for KNN

ROC Curve for ANN

The ROC Curve for ANN is depicted in the figure below.

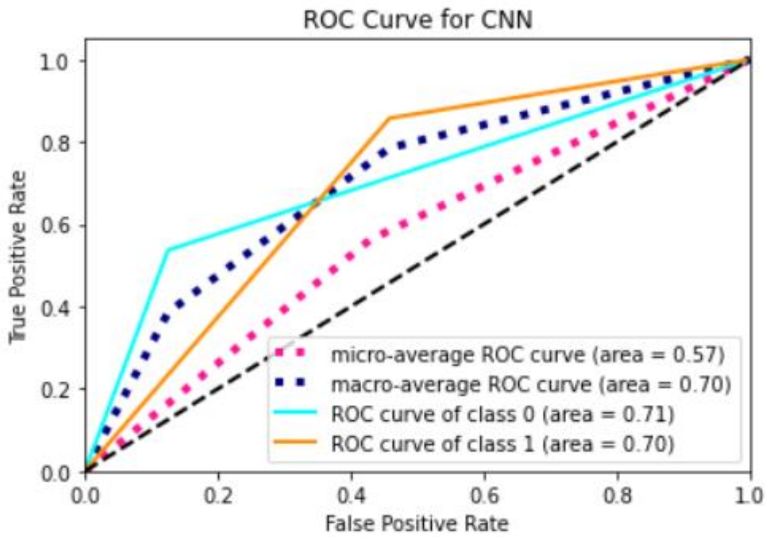

Fig. 9.7 ROC Curve for ANN

Confusion Matrix for ANN

The ANN Confusion Matrix is depicted in the figure below.

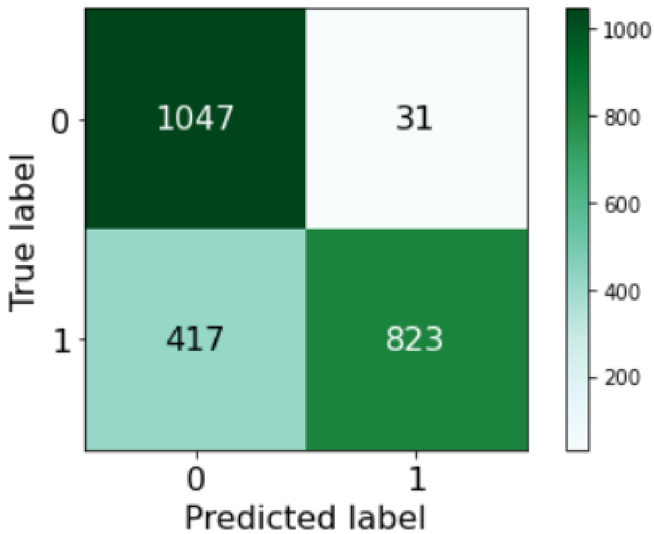

Fig. 9.8 Confusion Matrix for ANN

ROC Curve for Decision Tree

The ROC Curve for Decision Tree is depicted in the figure below.

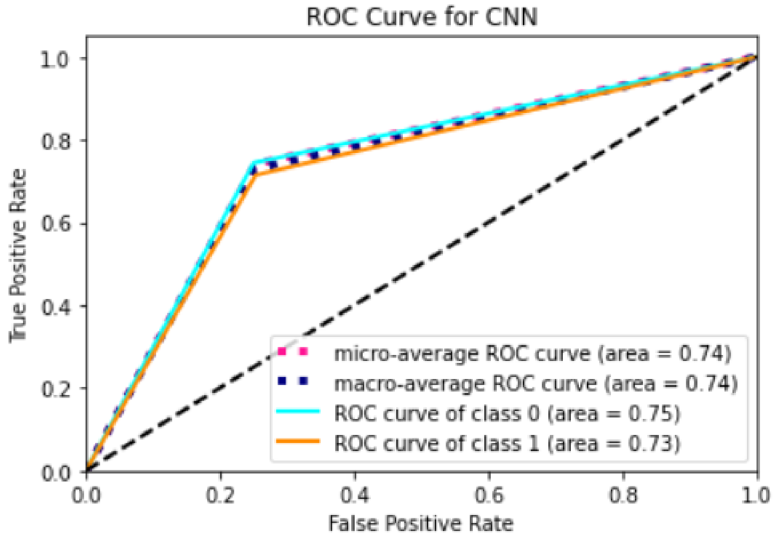

Fig. 9.9 ROC Curve for Decision Tree

Confusion Matrix for Decision Tree

The Decision Tree Confusion Matrix is depicted in the figure below.

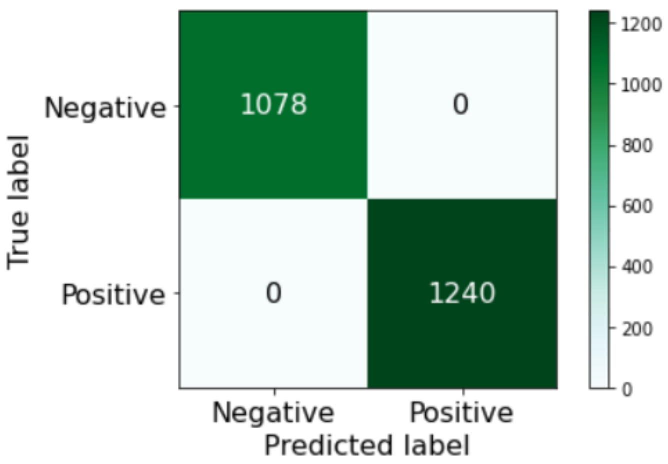

Fig. 9.10 Confusion Matrix for Decision Tree

Confusion Matrix

The confusion matrix is a machine learning predictive analysis tool. The confusion matrix is used to evaluate the performance of a classification-based machine learning model. Confusion matrix is also a table that summarises the number of correct and incorrect predictions made by a classifier (or a classification model) for binary classification tasks.

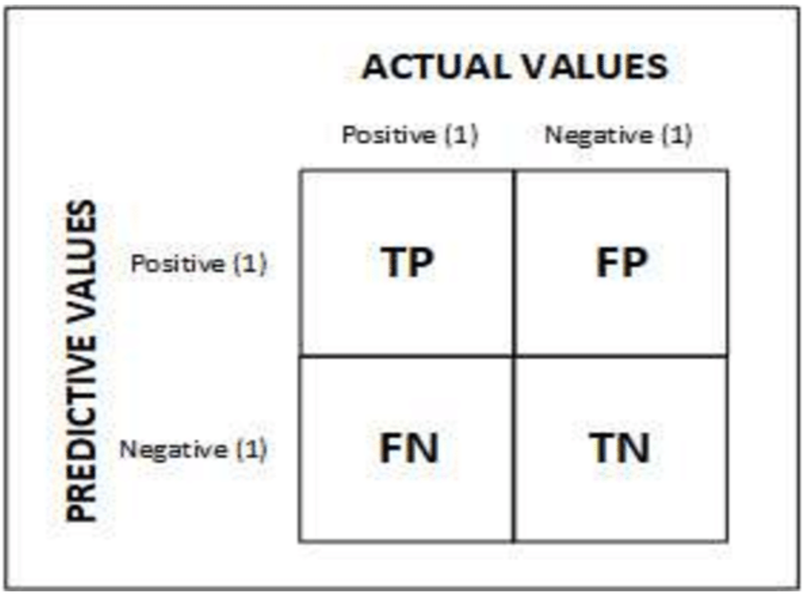

Fig. 9.11 Confusion Matrix

- **True Positive (TP):** The number of correctly predicted positive instances.
- **True Negative (TN):** The number of correctly predicted negative instances.
- **False Positive (FP):** The number of instances that were predicted as positive but were actually negative.
- **False Negative (FN):** The number of instances that were predicted as negative but were actually positive.

Accuracy

Accuracy is a measure of how many correct predictions your model made across the entire test dataset. Accuracy is a good starting point for measuring the model's performance. Accuracy becomes a poor metric in unbalanced datasets.

$$\textbf{Accuracy} = \frac{(\textbf{TP} + \textbf{TN})}{(\textbf{TP} + \textbf{TN} + \textbf{FP} + \textbf{FN})}$$

Precision

Precision signifies how many of the correctly predicted cases were positive. This would determine whether or not our model is reliable. Precision is a useful metric when false positives are more concerning than false negatives.

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}}$$

Recall(sensitivity)

Recall indicates how many of the actual positive cases our model correctly predicted. In cases where False Negative trumps False Positive, recall is a useful metric.

$$\mathbf{Sensitivity} = \frac{\mathbf{TP}}{\mathbf{TP + FN}}$$

The results are obtained when testing data is given for prediction. The results are evaluated using different metrices like accuracy. The prediction results of all the algorithms are compared and the best model is proposed. In proposed algorithms, the classification accuracy obtained by YOLO-CNN is 90-94%, Random Forest is 80%, Decision Tree is 73.3%, KNN is 55.6% and ANN is 55.6%. Among all the algorithms, YOLO-CNN obtained best results.

S.No	Model	Accuracy
1.	YOLO-CNN	94-96
2.	Random Forest	86
3.	Decision Tree	75
4.	KNN	73
5.	ANN	72

10.CONCLUSION

The use of YOLO and CNN in brain tumour detection and classification has shown promising results. YOLO can accurately locate the tumour within a medical image, while CNN can classify the tumour into different categories based on its characteristics like size of tumour.

The approach can potentially aid doctors in making more accurate and efficient diagnoses, leading to earlier detection and treatment of brain tumours. However, further research and testing is required to validate the effectiveness and practicality of this method. Overall, the combination of YOLO and CNN in brain tumour detection and classification has the potential to improve patient outcomes and ultimately save lives.

11.REFERENCES

1) Amran Hossain, Mohammad Tariqul Islam, Mohammad Shahidul Islam, Muhammed E. H. Chowdhury, ALI F. Almutairi, Norbahiah Misran,” A YOLOv3 Deep Neural Network Model to Detect Brain Tumour in Portable Electromagnetic Imaging System”. Published in 2021 Digital Object Identifier 10.1109/ACCESS.2021.3086624
2) Sethuram Rao.Grish, Vydeki.D,” Brain Tumour Detection Approaches: A Review using Artificial Neural Network (ANN) and Support Vector Machine (SVM)”.Published in 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT).
3) Sunil Kumar, Renu Dhir, Nisha Chaurasia ,” Brain Tumour Detection Analysis Using CNN: A Review using CNN classification technique and has been used to disregard the dataset picture algorithm error”. Published in International Conference on Artificial Intelligence and Smart Systems (ICAIS 2021)
4) Mohammad Omid Khairandish, Meenakshi Sharma, Kusrini Kusrini,” The Performance of Brain Tumour Diagnosis Based on Machine Learning Techniques Evaluation - A Systematic Review using CNN, KNN, C-means, RF, respectively, ordered from the highest frequency of use to the lowest. Published 2020 3rd International Conference on Information and Communications Technology (ICOIACT)
5) Parveen, Amritpal Singh,” Detection of brain tumour in MRI images, using combination of fuzzy c-means and SVM’, Published 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)
6) Mohammad Shahjahan Majib, Md Mahbubur Rahman, T. M. Shahriar Sazzad, Nafiz Imtiaz Khan, Samrat Kumar Dey “VGG-SCNet: A VGG Net-Based Deep Learning Framework for Brain Tumour Detection on MRI Images”. Published in 2021 IEEE Access (Volume: 9)
7) M.O. Khairandish, M. Sharma, V. Jain , J.M. Chatterjee, N.Z. Jhanjhi,” A Hybrid CNN-SVM Threshold Segmentation Approach for Tumour Detection and Classification of MRI Brain Images”.Published in IRBM Volume 43, Issue 4, August 2022, Pages 290-299
8) Simon Podnar, Matjaž Kukar, Gregor Gunčar, Mateja Notar, Nina Gošnjak, Marko Notar,” Diagnosing brain tumours by routine blood tests using machine learning”. Published in 2019 https://www.nature.com/articles/s41598-019-5114

12.APPENDIX

yolov4brain.m

Code for Training Data

Editor - C:\Users\vinay\OneDrive\Desktop\Project\brain yolo\yoloV4brain.m

yoloV4brain.m

```
%% Yolov_4 Brain tumor classification using CNN
clc
clear all
close all
imds = imageDatastore('C:\Users\MEBIN\Desktop\fundusmathworks\Database',...
    'IncludeSubfolders',true,...
    'LabelSource','foldernames');
[Data,testData]= splitEachLabel(imds,0.8,'randomize');%% allocating the labels for split data
%CNN Training files
[trainData] =Data;%% Training data
layers = [
    imageInputLayer([200 128 3],'Name','input')%%size of the input layer
    convolution2dLayer(5,16,'Padding','same','Name','conv_1')
    batchNormalizationLayer('Name','BN_1')
    reluLayer('Name','relu_1')
    convolution2dLayer(3,32,'Padding','same','Stride',2,'Name','conv_2')
    batchNormalizationLayer('Name','BN_2')
    reluLayer('Name','relu_2')
    convolution2dLayer(3,32,'Padding','same','Name','conv_3')
    batchNormalizationLayer('Name','BN_3')
    reluLayer('Name','relu_3')
    additionLayer(2,'Name','add')
    averagePooling2dLayer(2,'Stride',2,'Name','avpool')
    fullyConnectedLayer(2,'Name','fc')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classOutput')];
% Create a layer graph from the layer array. layerGraph connects all the layers in layers sequentially. Plot the layer graph.
lgraph = layerGraph(layers);
figure
plot(lgraph)
% Create the 1-by-1 convolutional layer and add it to the layer graph. Specify the number of convolutional filters and the stride so
skipConv = convolution2dLayer(2,32,'Stride',2,'Name','skipConv');
lgraph = addLayers(lgraph,skipConv);
figure
plot(lgraph)
% Create the shortcut connection from the 'relu_1' layer to the 'add' layer.
% Because you specified two as the number of inputs to the addition layer when you created it,
% the layer has two inputs named 'in1' and 'in2'. The 'relu_3' layer is already connected to the 'in1' input. Connect the 'relu_1' la
lgraph = connectLayers(lgraph,'relu_1','skipConv');
lgraph = connectLayers(lgraph,'skipConv','add/in2');
figure
plot(lgraph);
options = trainingOptions('adam', ...
    'MiniBatchSize',128, ...
    'MaxEpochs',2, ... % was 6
    'ValidationFrequency',5, ...
    'InitialLearnRate',1e-4,'Plots','Training-progress');
```

Code for Training Network Model

```
%% network training
 [convnet, traininfo] = trainNetwork(trainData,lgraph,options);
% INPUT IMAGE
[filename,pathname] = uigetfile({'*.*';'*.bmp';'*.tif';'*.gif';'*.png'},'Pick an Image File');
I = imread([pathname,filename]);
img1=rgb2gray(I);
img1 = imresize(img1,[200,128]);
% Converts uint8 to Double
img=im2double(img1);
%Removing Noise by applying Median Filter
 img=medfilt2(img);
imshow(img);
img1 = I;
if(size(img1,3)>2)
img1=rgb2gray(img1);
end
img1 = imresize(img1,[200,128]);
% Converts uint8 to Double
img=im2double(img1);
%Removing Noise by applying Median Filter
img=medfilt2(img);
bw = im2bw(img,0.7);
imgCheck = im2bw(img1);
% Crearing 2D Binary Image Label
label = bwlabel(bw);
% For Finding the Properties of the image
% Returns a scalar specifying the proportion of the pixels in the convex hull that are also in the region
properties = regionprops(label,'Solidity','Area');
% Density of the convex Hull
density = [properties.Solidity];
% Area of the Hull
area = [properties.Area];
% High Density Areas are   tumor
denseCond = density > 0.5;
% Finding the biggest tumor
denseMax = max(area(denseCond));
tumorFinal = find(area == denseMax);
% "configureYOLOv4" function to create a custom YOLO v4 network using the
% pretrained model.
```

Code for Importing the Trained Model

```
%% Setup
% Add path to the source directory.
addpath('src');
```

```
%% Download Pretrained Network
% This repository uses two variants of YOLO v4 models.
% *YOLOv4-coco*
% *YOLOv4-tiny-coco*
% Set the modelName from the above ones to download that pretrained model.
modelName = 'YOLOv4-coco';
model = helper.downloadPretrainedYOLOv4(modelName);
net = model.net;
```

Code for Masking

```
%% Load Data
% one or two labeled instances of a vehicle. A small data set is useful for exploring
% the YOLO v4 training procedure, but in practice, more labeled images are needed
% to train a robust network.
%
% Unzip the vehicle images and load the vehicle ground truth data.
data = load('YOLOv4-coco.mat');
envCfg = coder.gpuEnvConfig('host');
envCfg.DeepLibTarget = 'cudnn';
envCfg.DeepCodegen = 1;
envCfg.Quiet = 1;

addpath test;% adding the test data of YOLO.V4
% From the properties finded out earlier, we found the biggest tumor
yolotumor = ismember(label,tumorFinal);
% Using Morphological Operation using square structural Element
se = strel('square',3);
 yolotumor = imdilate(yolotumor,se);
ref=yolotumor;
imshow(yolotumor);
%mask code

Tumor1=bwareafilt(yolotumor,1);
figure(6), imshow(Tumor1);
 [m n]=size(ref);
 figure(6), subplot(131), imshow(ref),title('Yolo detection 1');
 I1=imresize(I,[m n]);
 subplot(132), imshow(I1);title('Yolo detection 2');
  I1=rgb2gray(I1);
 for i=1:m
     for j=1:n
          if (ref(i,j)==0)
             I1(i,j)=0;
          end
     end
 end
subplot(133), imshow(I1);title('Yolo detection 3');
%using Yolov 4 cc analysis
[L Ne]=bwlabeln(ref);
for n=1:Ne
        [r,c] = find(L==n);
        n1=I1(min(r):max(r),min(c):max(c));
end
mask=zeros(size(I1));
figure(7), subplot(131), imshow(n1),title('Yolo detection 5');
subplot(132), imshow(mask),title('Yolo detection mask');

img1 = I;
if(size(img1,3)>2)
img1=rgb2gray(img1);
end
```

Code for Feature Extraction

```
% Performing the Principal Component Extraction for feature extraction
G = pca(dwtfeat);
%FEATURE EXTRACTION
g = graycomatrix(G);
statistics = graycoprops(g,'Contrast Correlation Energy Homogeneity');
Contrast = statistics.Contrast;
Correlation = statistics.Correlation;
Energy = statistics.Energy;
Homogeneity = statistics.Homogeneity;
Mean = mean2(G);
Standard_Deviation = std2(G);
Entropy = entropy(G);
RMS = mean2(rms(G));
%Skewness = skewness(img)
Variance = mean2(var(double(G)));
a = sum(double(G(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(G(:)));
Skewness = skewness(double(G(:)));
Accuracy = Homogeneity*100;
disp('Accuracy = ' );
disp(Accuracy );
% Inverse Difference Movement
m = size(G,1); % No. of Rows
n = size(G,2); % No. of Columns
diff = 0;
for i = 1:m
    for j = 1:n
        temp = G(i,j)./(1+(i-j).^2);
        diff = diff + temp;
    end
end
IDM = double(diff);
features = [Contrast,Correlation,Energy,Homogeneity, Mean, Standard_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness

% originalPhoto = imread('tumor1.png');
grayscalePhoto = rgb2gray(I);
figure,imshow(grayscalePhoto)
title('Grayscale Image');
% Apply a high pass filter to the image
hpf = [-1 -1 -1;-1 10 -1;-1 -1 -1];
hpfImage = imfilter(grayscalePhoto, hpf, 'same');
figure,imshow(hpfImage),title('HPF Image');

% Apply and display a median filtered image
medianImage = medfilt2(hpfImage, [3,3]);
figure
imshow(medianImage)
title('Median Image')

% Compute and display image opening threshold
se = strel('disk',20);
Io = imopen(medianImage,se);
figure,imshow(Io)
title('Yolo Foreground Image');
% Reconstruct the original image
Ie = imerode(medianImage, se);
Yolo = imreconstruct(Ie, medianImage);
figure
imshow(Yolo)
title('Yolo Reconstructed Image')
% Apply clarification to the reconstructed image
cl = medfilt2(Yolo, [5,5]);
figure,imshow(cl)
title('Yolo Clarified Image')
% Show current progress
imshowpair(grayscalePhoto,cl,'montage')
% Threshold the doctored image into 3 layers
yolothresh = multithresh(cl,2);
thresholdImage = imquantize(cl,yolothresh);
figure(2)
subplot(1,2,1)
imshow(thresholdImage,[])
title('Yolo Threshold image')
% Pick out the yolo top layer as the tumor
yolotumor = thresholdImage;
for i = 1:length(thresholdImage)
    for j = 1:length(thresholdImage)
%           if thresholdImage(i,j) < 3
            yolotumor(i,j) = 1;
        end
    end
```

Code for CNN classification

```
%% Done classification
    class = classify(convnet,I);
    msgbox(char(class))
msgbox('Classification of Brain Tumor using Yolo v_4 algorithm successfully tested using CNN');
```

23B05

ORIGINALITY REPORT

27%

SIMILARITY INDEX

PRIMARY SOURCES

1	www.cse.griet.ac.in Internet	104 words — 2%
2	ijiemr.org Internet	73 words — 2%
3	medium.com Internet	66 words — 1%
4	ozenero.com Internet	53 words — 1%
5	www.kevsbox.com Internet	53 words — 1%
6	www.section.io Internet	48 words — 1%
7	dokumen.tips Internet	47 words — 1%
8	Bolanle A. Ojokoh, Omotolani A. Olaku, Oluwafemi A. Sarumi, Samuel I. Olotu. "Predictive analytics for economic crisis triggered depression risk level identification among some adults in Nigeria", Scientific African, 2021 Crossref	41 words — 1%