Advanced Automated System for Detecting Brain Tumor Abnormalities in MRI

A Project Report
Submitted in the partial fulfillment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE ENGINNERING

By

G. GAGAN- 2320030115
M. SREE VATHSAV- 2320030078
J. CHIDRUPH -2320030107

Under the Esteemed Guidance of

VENKATESWARA RAO PULIPATI

(DEEMED TO BE UNIVERSITY)

Koneru Lakshmaiah Education Foundation
(Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Off-Campus: Bachupally-Gandimaisamma Road, Bowrampet, Hyderabad, Telangana - 500 043.
Phone No: 7815926816, www.klh.edu.in

K L (Deemed to be) University

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Declaration

The Project Report entitled “**CURE ML**“is a record of Bonafide work of **G.GAGAN-2320030115,team members M.SREE VATHSAV, J.CHIDRUPH** submitted in partial fulfillment for the award of B. Tech in Computer Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

G. GAGAN-2320030115

M.SREE VATHSAV-2320030078

J.CHIDRUPH-2320030107

K L (Deemed to be) University

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Certificate

This is certify that the project based report entitled “**CURE ML** ” is a bonafide work done and submitted by **G.GAGAN (2320030115),team members M.SREE VATHSAV(2320030078), J.CHIDRUPH(2320030107)** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in Department of Computer Science Engineering, K L (Deemed to be University), during the academic year **2024-2025.**

Signature of the Supervisor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our wish to express my sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my guide, **VENKATESWARA RAO PULIPATI** Department of Computer Science which cannot be expressed in words for his tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. RAMESH BADU**, Head of the Department for Computer Science Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project based Lab.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project. Finally, I sincerely thank my parents, friends and classmates for their kind help and cooperation during my work.

G. GAGAN-2320030115

M.SREE VATHSAV-2320030078

J.CHIDRUPH-2320030107

TABLE OF CONTENTS

S.No	Contents	Page no
1	Abstract	7
2	Introduction	8
3	Literature survey	9
4	Client meetings	10
5	Hardware and Software requirements	15
6	Implementation	17
7	Experimentation and Code	18
8	Results	23
9	Conclusion	25
10	References	26

ABSTRACT

TITLE:

Advanced Automated System for Detecting Brain Tumor Abnormalities in MRI Using machine learning

BACKGROUND: Accurate interpretation of MRI scans is essential for diagnosing and managing a wide range of medical conditions. Traditional methods of image analysis rely heavily on the expertise of radiologists, which can lead to variability in diagnostic outcomes and increased risk of error. There is a growing need for automated systems that can enhance diagnostic precision and efficiency in medical imaging.

OBJECTIVE: The objective of this project is to develop an advanced automated system that employs machine learning techniques to detect abnormalities in MRI scans. The system aims to provide radiologists with a reliable, efficient, and accurate tool for improving diagnostic accuracy and accelerating workflow.

METHODS:

1.DATA ACQUISITION AND PREPARATION

Dataset: Utilizes pituitary tumor MRI can images that publicly available dataset form Kaggle for brain tumor These datasets are annotated by expert radiologists and provide a diverse range of abnormalities.

Preprocessing: Images are preprocessed to reduce noise, normalize intensity, and augment data using techniques such as rotation, scaling, and translation to improve model robustness.

2. MACHINE LEARNING MODEL DEVELOPMENT:

Algorithms:

Support Vector Machines (SVM): Employed for classification by finding the optimal hyperplane that separates the data into different classes. Different kernels such as linear, radial basis function (RBF), and polynomial are evaluated for the best classification performance.

Feature Extraction: Features are extracted from the images using methods like Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) to reduce the dimensionality before feeding them into the SVM for classification.

Training: The model is trained using a supervised learning approach with a training dataset and validated using a separate validation dataset. Cross-validation techniques are applied to ensure the model generalizes well to unseen data, and hyperparameters like the regularization parameter (C) and kernel parameters (gamma, degree) are tuned for optimal performance.

SYSTEM INTEGRATION:

User Interface: A user-friendly interface is designed for radiologists to interact with the system. Features include automated anomaly detection and visualization of detected abnormalities.

Tools Used: Python programming with libraries such as numpy , pandas ,sklearn for model development; OpenCV for image processing; streamlit for web-based interface development.

PERFORMANCE EVALUATION:

Metrics: Evaluated using accuracy, sensitivity and specificity.

Validation: Cross-validation is performed to assess the model's effectiveness.

CONTINUOUS LEARNING AND IMPROVEMENT:

Feedback Mechanism: Integrated to refine the model based on real-world usage and user feedback. This includes periodic retraining with new data and adapting to evolving diagnostic standards.

Introduction

In the field of healthcare, particularly medical imaging, the timely and accurate diagnosis of critical conditions like brain tumors is essential to improving patient outcomes. Magnetic Resonance Imaging (MRI) is widely used for brain imaging, but the traditional process of interpreting these scans is heavily reliant on the expertise of radiologists. This manual approach can be time-consuming, prone to variability, and subject to human error, especially as the volume of scans continues to increase. To address these challenges, our project focuses on developing an **Advanced Automated System for Detecting Brain Tumor Abnormalities in MRI Scans Using Machine Learning**. By leveraging artificial intelligence (AI), specifically machine learning (ML) techniques, our system aims to enhance the diagnostic accuracy and efficiency of brain tumor detection.

The project uses publicly available datasets of MRI scans, focusing on pituitary tumors, and applies machine learning models to automate the identification of abnormalities. The model development includes data preprocessing, feature extraction using methods like Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT), and classification using Support Vector Machines (SVM) with various kernel functions. Additionally, a user-friendly interface is designed using Streamlit to enable radiologists to interact with the system seamlessly, facilitating quicker decision-making.

The primary goal is to reduce the workload on radiologists by providing a reliable and efficient tool for brain tumor detection, ultimately leading to faster diagnoses and improved patient care. This system not only accelerates the diagnostic process but also ensures consistency and scalability, making it a valuable asset in modern healthcare settings. By integrating continuous learning mechanisms, our project ensures that the system evolves with new data, enhancing its accuracy and adaptability over time. Through this innovative solution, we aim to revolutionize the way radiologists interpret MRI scans, aligning with global healthcare goals of improving diagnostic outcomes and patient well-being

Literature survey

Deciding to implement machine learning in healthcare, particularly for the detection of a brain tumour, would be something I never did lightly. I am afraid, first and foremost, of introducing an ML system into any sensitive area such as medical diagnostics; there are too many challenges that process entailed-a mix of the ethical, practical, and technical. My first concern would be what happened with automated systems in a field involving human life. Healthcare incorrect diagnoses, errors, or false positives can lead to grave consequences. Hence, most experts argue against using AI and ML, using this argument without seriously considering the issue. There are problems of privacy in relation to patient data, potential algorithmic bias, and the need for interpretability, which make healthcare a very challenging domain for ML applications. However, I came to realize that these issues only made me think that even deeper research and development had to be made to overcome such difficulties, rather than discouraging people from using ML.

With all these in mind, I decided to dig a bit deeper in the study of how ML interfaced with diagnostics regarding brain tumors. Extensive research concerning the prevalence and complexities of brain tumors is quite intensive, especially concerning the less easy-to-diagnose early pituitary tumors with subliminal and variable presentations on MRI scans. It was found that detection of the tumor earlier and accurately greatly enhances the outcomes of patients. I went about various academic sources, scientific journals, and online mediums, which allowed me to build the comprehensive knowledge of what the available diagnostic technologies can or cannot do. I discovered how radiologists typically interpret the MRI images and the limitations in manual analysis- human mistakes, fatigue, and disparity in the diagnosis.

But I did not end there with just online research alone. To get a real-world perspective, I went straight to the healthcare professionals involved. I took time to visit hospitals, MRI scan rooms, and clinics and had the opportunity to present this to radiologists and other medical staff who are directly involved with the diagnosis and treatment of these patients. Through these interactions, I directly learned about the diagnostic process, challenges healthcare providers face, and where automation can ease matters rather than replace human expertise. The radiologists provided examples of their experiences and those underlining heavy pressures on them in case of accurate diagnoses while dealing with increased workloads. They also mentioned the inadequacies in the tools so far developed for specific tumor types and an open obligation on systems that should be able to help them instead of creating another hurdle for them in their diagnostic procedures.

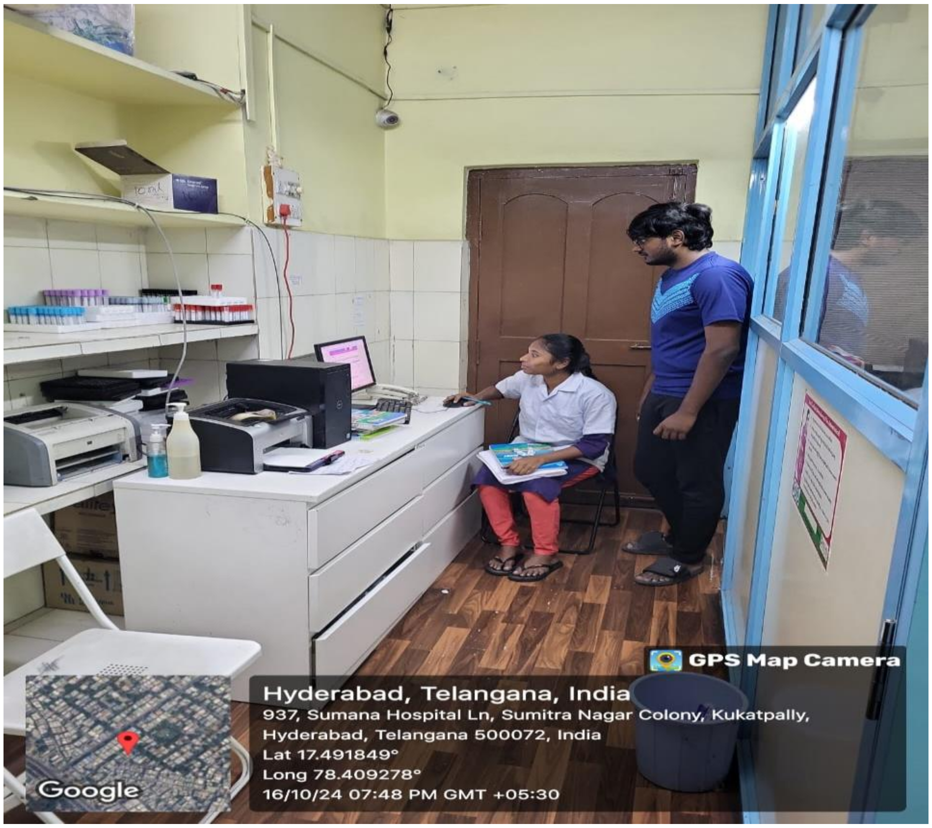
I wasn't just on an expedition of research with academic studies and runs to hospitals; it was more of a combination approach which includes theory and practice. I tried to fill this gap through direct interaction with scientific literature and the medical community, thus bringing in a hands-on approach to technological potential and real-world applicability. This was very helpful in understanding the subtlety involved in detection of brain tumors as well as the challenges to be addressed while building up an ML model for healthcare applications. In other words, it is based on a balanced knowledge of the theoretical framework and the practical reality in medical diagnostics.

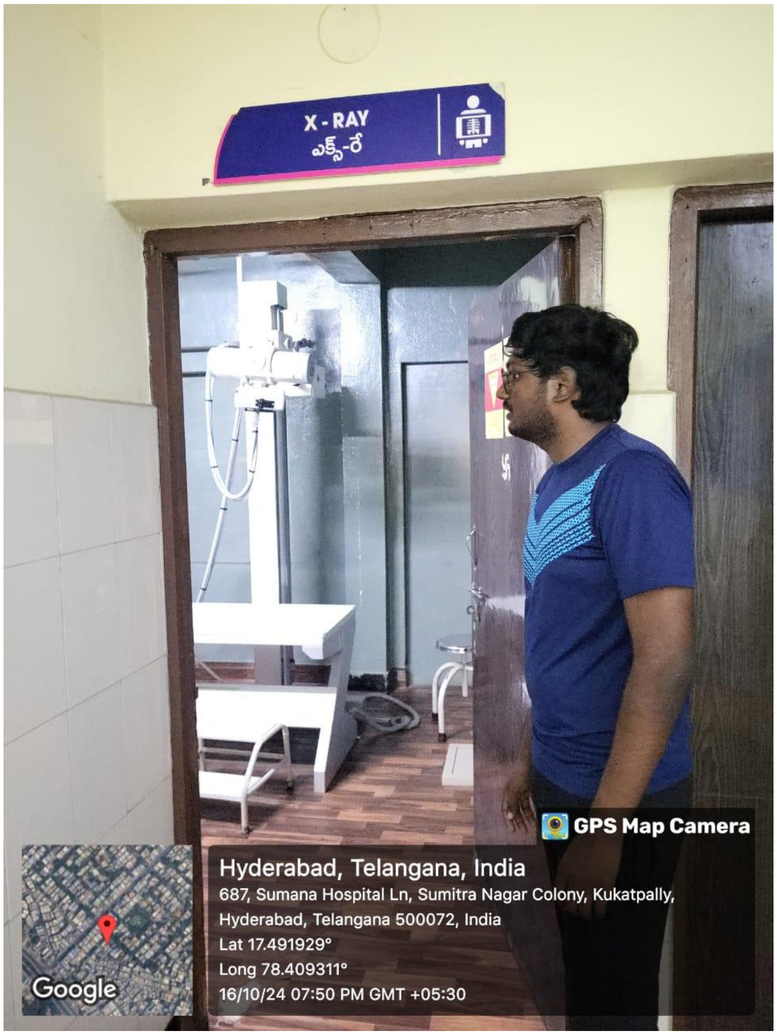
This project, therefore, is not merely an attempt to apply machine learning in some novel way but is, in fact, a proper, well-thought initiative that emerged from long research and real-world observations at large. In integrating ML with health diagnostics, specifically with reference to brain tumors, I hope to create a resource that will help radiologists enhance the accuracy of diagnoses for the betterment of the patient's outcome. This view embodies the notion that though ML cannot replace human know-how blindly, it is most certainly to be a very effective assistant in the health care process when created with a deep understanding of the domain and an ethical commitment to patient safety.

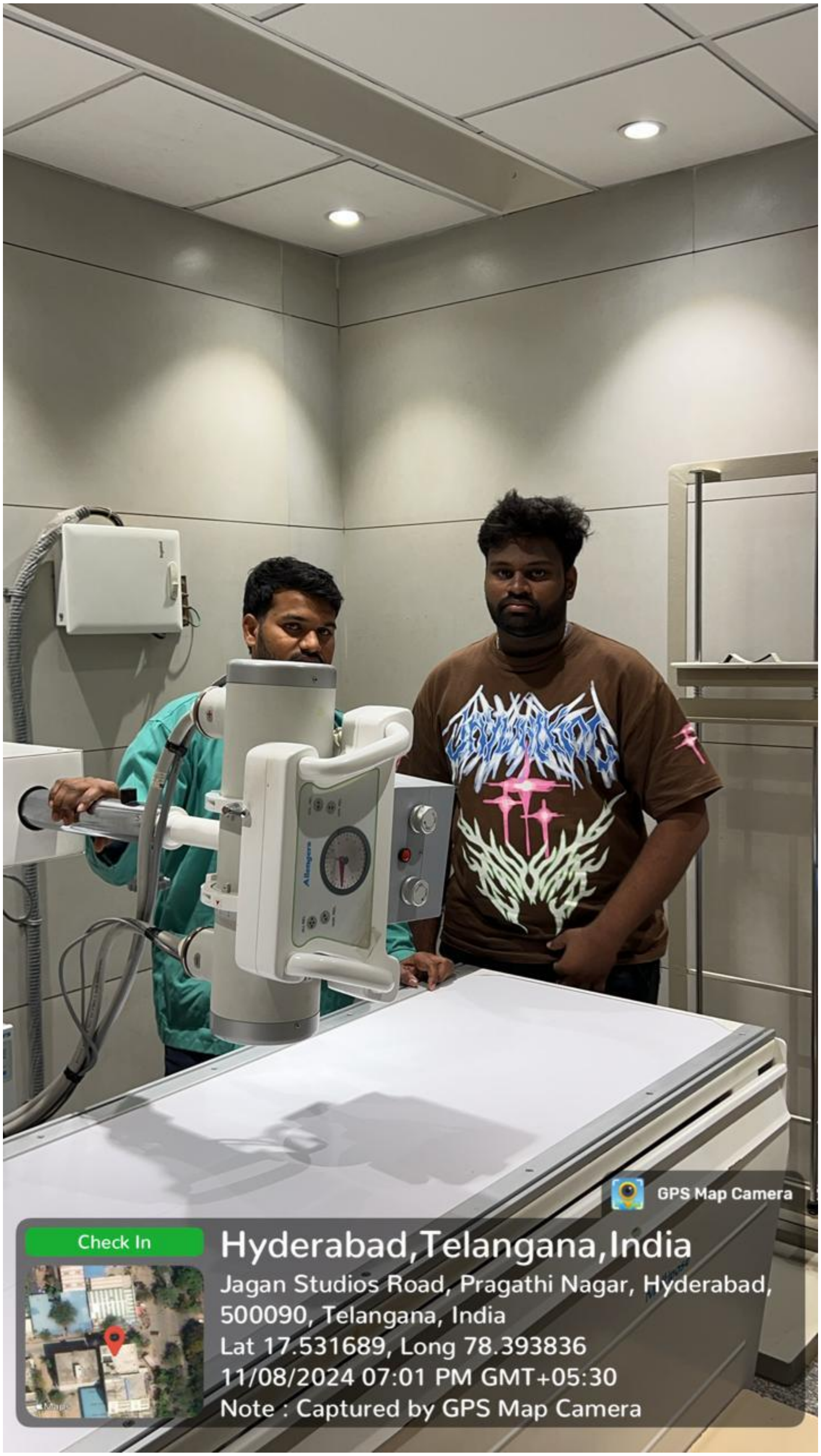
Client meetings

During visits made to the hospital and MRI scan rooms, I found that the rooms were well-organized with ecquipment. That reflects the experience of scanning made by the radiologists on the scanner by taking maximum precautions and interaction with the patients. Nurses run these clinics very efficiently in carrying even administrative tasks along with meeting the sensitive needs of patients at every step.

In the case of both radiologists and nurses, they indicated that they can indeed work appropriately together in the process toward improving their experiences as patients. Summing up their visit, there was a commitment to quality and patient-centered care through the teamwork they displayed and sensitivity to standards.


GPS Map Camera
Hyderabad, Telangana, India
937, Sumana Hospital Ln, Sumitra Nagar Colony, Kukatpally,
Hyderabad, Telangana 500072, India
Lat 17.491849°
Long 78.409278°
16/10/24 07:48 PM GMT +05:30
Google

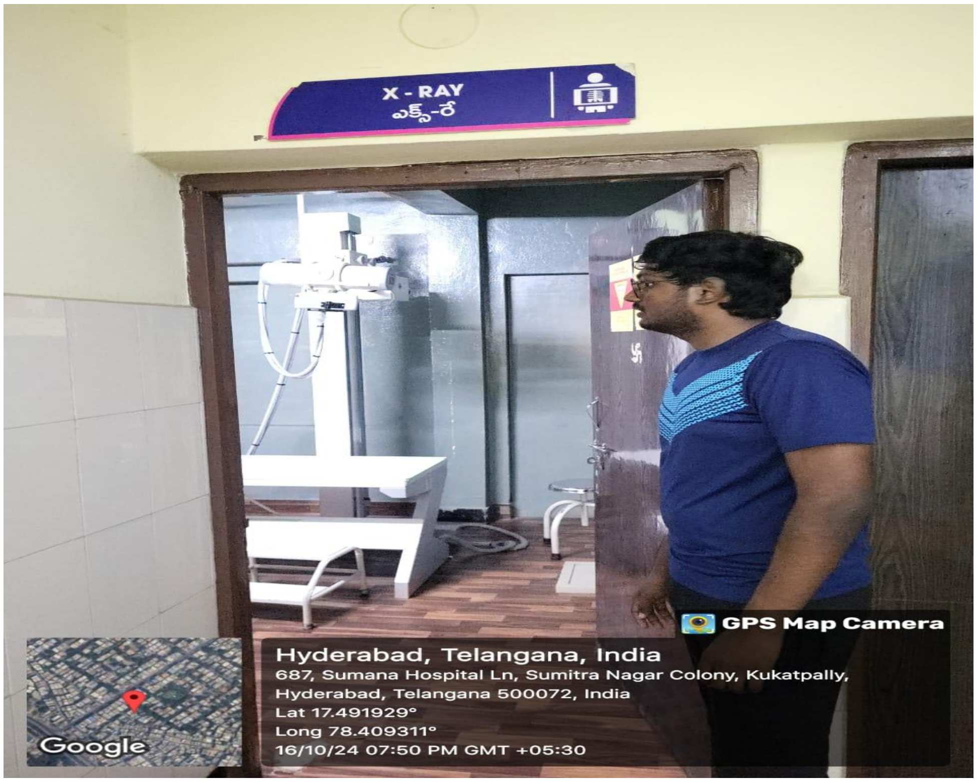

X - RAY
ఎక్స్-రే
GPS Map Camera
Google
Hyderabad, Telangana, India
687, Sumana Hospital Ln, Sumitra Nagar Colony, Kukatpally,
Hyderabad, Telangana 500072, India
Lat 17.491929°
Long 78.409311°
16/10/24 07:50 PM GMT +05:30



Hyderabad,Telangana,India
Jagan Studios Road, Pragathi Nagar, Hyderabad,
500090, Telangana, India
Lat 17.531689, Long 78.393836
11/08/2024 07:01 PM GMT+05:30
Note : Captured by GPS Map Camera

Hyderabad,Telangana,India

Jagan Studios Road, Pragathi Nagar, Hyderabad,
500090, Telangana, India
Lat 17.531809, Long 78.393816
11/08/2024 07:01 PM GMT+05:30
Note : Captured by GPS Map Camera


X - RAY
ఎక్స్-రే
GPS Map Camera
Hyderabad, Telangana, India
687, Sumana Hospital Ln, Sumitra Nagar Colony, Kukatpally,
Hyderabad, Telangana 500072, India
Lat 17.491929°
Long 78.409311°
16/10/24 07:50 PM GMT +05:30
Google

X - RAY
ఎక్స్-రే
Horizon
GPS Map Camera
Hyderabad, Telangana, India
687, Sumana Hospital Ln, Sumitra Nagar Colony, Kukatpally,
Hyderabad, Telangana 500072, India
Lat 17.491929°
Long 78.409311°
16/10/24 07:50 PM GMT +05:30
Google

Software requirements

For our brain tumor detection project using machine learning, we utilized various software tools and libraries to manage tasks like data preparation, model training, feature extraction, and creating an easy-to-use interface.

Below is an explanation of each software we used and how it contributed to the project.

1. Python 3.8+

Python served as the main programming language for this project because it is simple to use, versatile, and comes with a wide range of libraries tailored for machine learning and data science. It's perfect for healthcare projects because it allows for quick development and easy collaboration with professionals who may not be familiar with complex coding. Its extensive support for AI and data processing makes it ideal for handling MRI data efficiently.

2. Jupyter Notebook

We used Jupyter Notebook as our primary development environment. This tool allows you to write and run Python code in sections, making it easier to test models, analyze results, and document your work—all in one place.

It's particularly useful for visualizing results and explaining our process step by step. By combining code, charts, and notes, it helps make the project understandable to both technical and non-technical stakeholders, like radiologists.

3. Libraries and Frameworks

a) NumPy

NumPy is crucial for performing numerical calculations and handling large datasets, especially when working with MRI images, which are essentially grids of pixel intensities. It efficiently manages the mathematical operations needed to preprocess these images.

b) Pandas

Pandas is used for organizing, cleaning, and preparing the data. It makes it easy to handle datasets, clean up any inconsistencies, and get the data ready for analysis. This helps ensure that the data fed into the model is accurate and well-organized.

c) scikit-learn (sklearn)

scikit-learn is a powerful machine learning library that we used to build our classification models. In this project, we relied on Support Vector Machines (SVM) to differentiate between scans with tumors and those without. The library also includes tools for evaluating model performance and fine-tuning settings to get the best results.

d) OpenCV

OpenCV is an open-source computer vision library used for image processing. It played a key role in enhancing MRI images by reducing noise, adjusting brightness, and performing feature extraction. This was essential for making sure our model received high-quality data to learn from.

e) Matplotlib These visualization tools helped us create graphs and charts to understand the data better and showcase our model's performance. For example, we used them to generate confusion matrices and accuracy graphs, which are vital for assessing how well our model is doing.

f) Keras (for future extensions)

Although our initial approach uses SVMs, we may later explore using deep learning models with TensorFlow or Keras to enhance detection accuracy, especially for more complex MRI patterns.

4. Streamlit

Streamlit was used to create an interactive web interface for the project. It allows radiologists and healthcare professionals to upload MRI scans and get immediate results from the model. This tool is excellent for quickly building a user-friendly front-end without needing extensive web development knowledge.

It provides an intuitive interface where users can visualize the scan results, making it easier for doctors to interpret the findings.

5. MySQL

For storing patient data and scan results, we considered using databases like MySQL or MongoDB. These systems would allow us to securely store patient records, past MRI scans, and diagnostic results, which can be useful for tracking patient history and conducting further analysis.

6. Anaconda Distribution

To manage all the different libraries and dependencies, we used Anaconda. It simplifies installing Python packages and ensures that everything works smoothly together, preventing compatibility issues.

Anaconda also helps manage isolated environments so that changes made to one project don't interfere with others. This is especially helpful when dealing with complex projects involving many different libraries.

7. Integrated Development Environments (IDEs)

For writing and debugging our code, we used IDEs like **PyCharm**.

implementation

Step 1: Setup and Environment

To get started, I set up my workspace by installing PyCharm, an IDE that helped me manage all the files, libraries, and code efficiently. Next, I downloaded the essential libraries needed for the project. These included numpy and pandas for handling data, sklearn for preprocessing, training, and evaluating models, opencv (or PIL) for image processing, and visualization tools like matplotlib and seaborn.

Step 2: Data Preparation

For the data, I gathered a collection of images classified into two categories: *No Tumor* and *Pituitary Tumor*. Each image needed to be uniform in size, so I resized and reshaped them to a standard pixel dimension. Then, I divided this data into training and testing sets, typically with an 80-20 or 70-30 split, to ensure that the model would have enough data to learn from while still reserving some for testing its accuracy.

Step 3: Feature Scaling (Considered but Skipped)

At this stage, I considered applying feature scaling to standardize pixel values. Feature scaling can often improve model performance, but I decided not to implement it in this initial setup.

Step 4: Model Training

With the data ready, I began training the models. First, I applied logistic regression, a straightforward classification algorithm, to establish a baseline performance. After that, I used Support Vector Machine (SVM), a more advanced model particularly well-suited for image classification. SVM leverages margin-based classification, which often enhances accuracy for distinguishing between two classes, such as “No Tumor” and “Tumor.”

Step 5: Model Evaluation

Once the models were trained, I evaluated their accuracy on the test set. This step was crucial to determine how well each model could differentiate between images with and without a tumor. Beyond accuracy, I also considered metrics like precision, recall, and the F1-score to get a well-rounded understanding of each model’s performance.

Step 6: Output Results

Finally, I examined the testing output. For a more intuitive presentation, I displayed sample images from the test set along with their predicted labels (“Tumor” or “No Tumor”). This visual approach not only confirmed the model’s accuracy but also made it easier to interpret and explain the results effectively.

Experimentation and Code

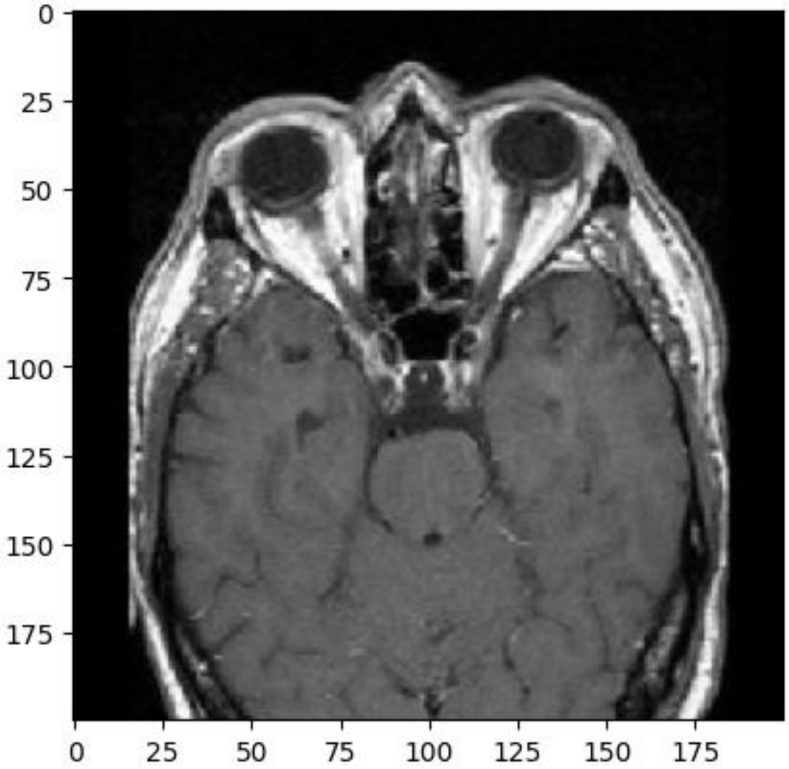
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score


import os

path = os.listdir('brain_tumor/Training/')
classes = {'no_tumor':0, 'pituitary_tumor':1}

import cv2
X = []
Y = []
for cls in classes:
    pth = 'brain_tumor/Training/'+cls
    for j in os.listdir(pth):
        img = cv2.imread(pth+'/'+j, 0)
        img = cv2.resize(img, (200,200))
        X.append(img)
        Y.append(classes[cls])

X = np.array(X)
Y = np.array(Y)

X_updated = X.reshape(len(X), -1)
```

```
np.unique(Y)
```

```
pd.Series(Y).value_counts()
```

```
X.shape, X_updated.shape
```

```
plt.imshow(X[0], cmap='gray')
```



```
X_updated = X.reshape(len(X), -1)
X_updated.shape
```

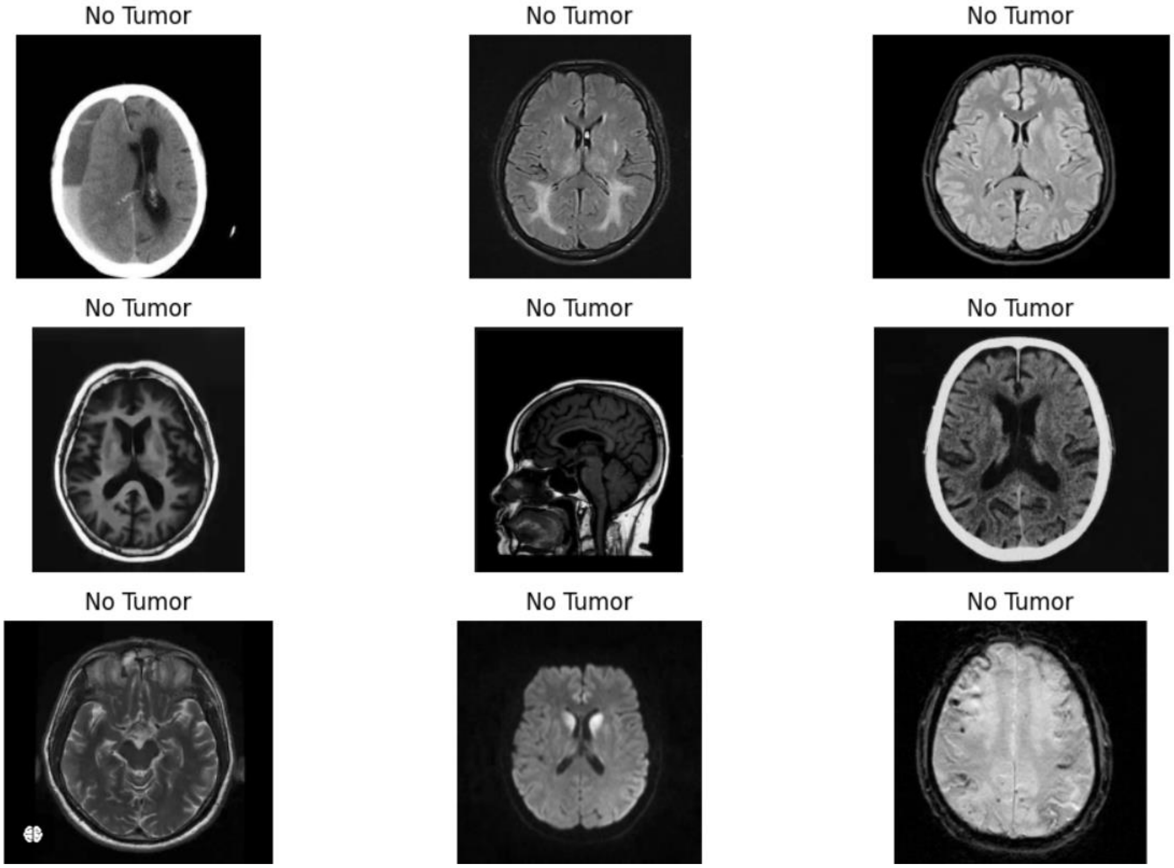
```
xtrain, xtest, ytrain, ytest = train_test_split(X_updated, Y, random_state=10,
                                               test_size=.20)
```

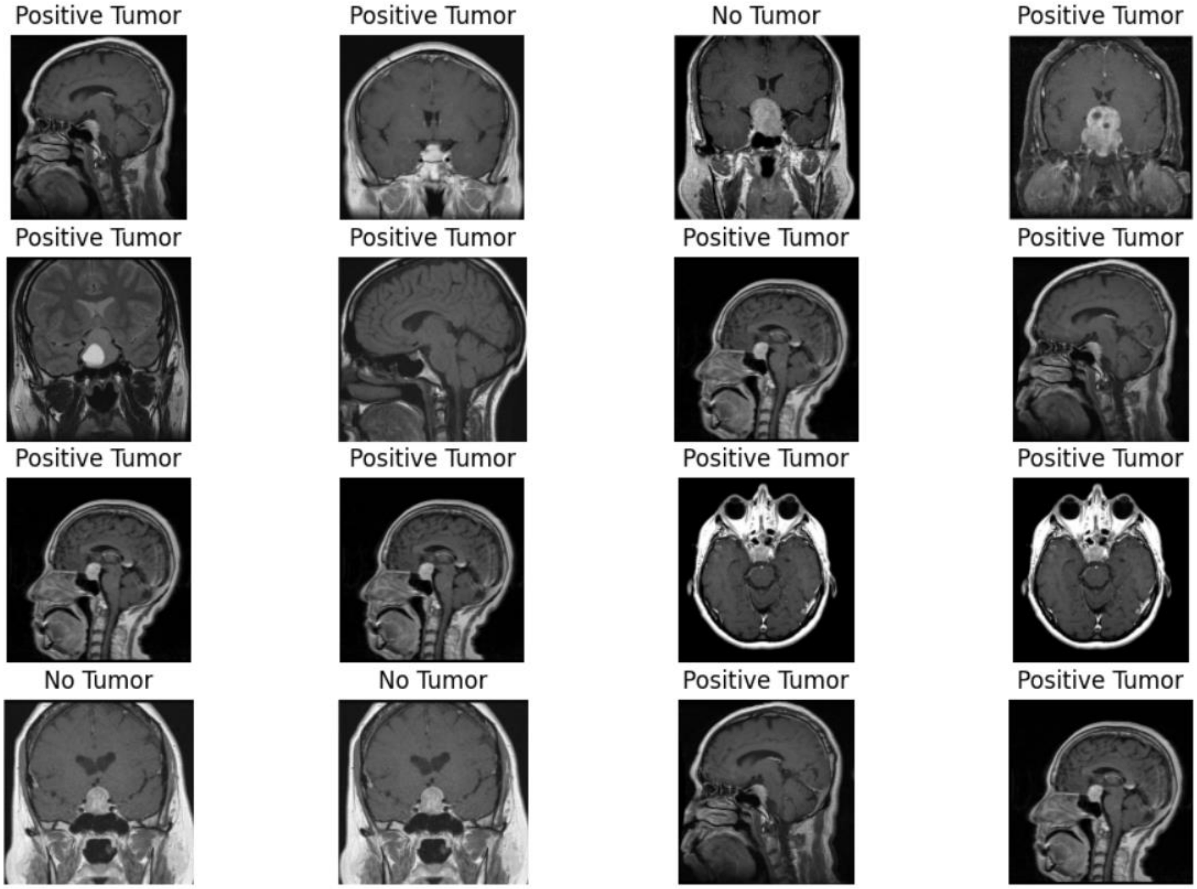
```
xtrain.shape, xtest.shape
```

```
print(xtrain.max(), xtrain.min())
print(xtest.max(), xtest.min())
xtrain = xtrain/255
xtest = xtest/255
print(xtrain.max(), xtrain.min())
print(xtest.max(), xtest.min())
from sklearn.decomposition import PCA

print(xtrain.shape, xtest.shape)

pca = PCA(.98)
# pca_train = pca.fit_transform(xtrain)
# pca_test = pca.transform(xtest)
pca_train = xtrain
pca_test = xtest

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC

mport warnings
warnings.filterwarnings('ignore')

lg = LogisticRegression(C=0.1)
lg.fit(xtrain, ytrain)

sv = SVC()
sv.fit(xtrain, ytrain)
```

```
print("Training Score:", lg.score(xtrain, ytrain))
print("Testing Score:", lg.score(xtest, ytest))

print("Training Score:", sv.score(xtrain, ytrain))
print("Testing Score:", sv.score(xtest, ytest))

pred = sv.predict(xtest)

misclassified=np.where(ytest!=pred)
misclassified

print("Total Misclassified Samples: ",len(misclassified[0]))
print(pred[36],ytest[36])

dec = {0:'No Tumor', 1:'Positive Tumor'}

plt.figure(figsize=(12,8))
p = os.listdir('brain_tumor/Testing/')
c=1
for i in os.listdir('brain_tumor/Testing/no_tumor/')[:9]:
    plt.subplot(3,3,c)

    img = cv2.imread('brain_tumor/Testing/no_tumor/'+i,0)
    img1 = cv2.resize(img, (200,200))
    img1 = img1.reshape(1,-1)/255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(img, cmap='gray')
    plt.axis('off')
    c+=1



plt.figure(figsize=(12,8))
p = os.listdir('brain_tumor/Testing/')
c=1
for i in os.listdir('brain_tumor/Testing/pituitary_tumor/')[:16]:
    plt.subplot(4,4,c)

    img = cv2.imread('brain_tumor/Testing/pituitary_tumor/'+i,0)
    img1 = cv2.resize(img, (200,200))
    img1 = img1.reshape(1,-1)/255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(img, cmap='gray')
    plt.axis('off')
    c+=1
```

RESULT-


No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor


Positive Tumor
Positive Tumor
No Tumor
Positive Tumor
Positive Tumor
Positive Tumor
Positive Tumor
Positive Tumor
Positive Tumor
Positive Tumor
Positive Tumor
Positive Tumor
No Tumor
No Tumor
Positive Tumor
Positive Tumor

Conclusion-

In this project, we developed a machine learning system to detect brain tumors using MRI data, comparing the performance of Logistic Regression and Support Vector Machine (SVM) models. Our results showed that SVM achieved higher accuracy than Logistic Regression due to its ability to capture complex patterns in the data. This highlights SVM’s potential in accurately classifying MRI scans, making it a valuable tool to assist radiologists in diagnosis. Future work could explore deep learning models like CNNs to further enhance detection accuracy and automate the process for improved patient outcomes.

References

Kaggle Dataset: Used for obtaining MRI scan data of brain tumors

Medical Research Papers: Provided insights into MRI-based tumor diagnosis

Logistic Regression /Support Vector Machines (SVM) learnt and Referenced for understanding SVM theory and application in classifying medical images.