A

Mini-Project Report on

Brain Tumour Detection using CNN

Submitted in partial fulfilment of the requirements of the degree of

BACHELOR OF ENGINEERING

IN

Computer Science & Engineering

Artificial Intelligence & Machine Learning

by

Mihir Rajendra Manjrekar	21106027
Rohit Pramod Negi	21106053
Shubham Shubhash Singh	21106030
Nehali Santosh Palkar	21106057

Under the guidance of

Dr. Jaya Gupta

Department of Computer Science & Engineering

Artificial Intelligence & Machine Learning

A. P. Shah Institute of Technology

G.B. Road, Kasarvadavli, Thane (W) - 400615

University of Mumbai

(2022 - 2023)

CERTIFICATE

This is to certify that the Mini Project A entitled "**Brain Tumour Detection using CNN**" is a bonafide work of **"Mihir Manjrekar (21106027), Rohit Negi (21106053), Shubham Singh (21106030), Nehali Palkar (21106057)"** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

__________________	__________________
Project Guide	Head of Department
Dr. Jaya Gupta	Dr. Jaya Gupta

Project Report Approval for S.E.

This Mini project report entitled ***Brain Tumour Detection using CNN*** by ***Mihir Manjrekar, Rohit Negi, Shubham Singh and Nehali Palkar*** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science & Engineering (Artificial Intelligence and Machine Learning) 2022 – 2023.***

Examiner Name Signature 1. ______________________

2. ______________________

Date:

Place:

Declaration

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

Mihir R. Manjrekar

Rohit P. Negi

Shubham S. Singh

Nehali S. Palkar

Date:

Abstract

Brain tumours are a serious medical condition that requires timely and accurate diagnosis for effective treatment. Medical imaging, such as magnetic resonance imaging (MRI) and computed tomography (CT) scans, are commonly used for brain tumour detection. However, manual interpretation of these images can be time-consuming and subject to human error. Therefore, developing an automated and accurate brain tumour detection system using machine learning techniques has gained significant attention in recent years.

The main objective of our project is to detect brain tumour by using Convolutional Neural Network(CNN) and VGG16. This project aims to build a machine learning model for brain tumour detection from images. A labelled dataset of brain MRI or CT images will be used for model training and evaluation. Various machine learning algorithms, such as convolutional neural networks (CNNs), Transfer Learning (using VGG-16) will be explored to develop a robust and accurate model. The dataset will be preprocessed to normalise pixel values, and data augmentation techniques may be applied to increase the diversity and size of the dataset.

The performance of the developed model will be evaluated using metrics such as accuracy, precision, recall, and F1 score. Cross-validation and/or a separate validation dataset will be used to assess the model's ability to generalise to unseen data. Hyperparameter tuning and model architecture adjustments will be performed to optimise the model's performance. The project's findings can benefit the field of medical imaging and machine learning, advancing the development of effective and reliable tools for brain tumour detection.

Table of Contents

List of Figures and Tables

Figures

Tables

CHAPTER 1

Introduction

1.1. Problem Definition

Cancer treatment technology has come a long way. Statistics from the National Cancer Institute show that there has been an overall decline in the cancer death rate. [1] However, this does not mean that we have conquered cancer. Even today, more than "50% cancer deaths" [2] happen due to late diagnosis. Patients and doctors are unable to correctly identify the symptoms of cancer. This can be attributed to many factors including but not limited to human error, limitations of manual processing, and risks involved in invasive diagnostic tests. However, with the advent of machine learning and advancements in classification techniques, hours of manual diagnostic work can be automated with net benefits.

1.2. Objectives

Our project aims to develop a model and an interface that allows medical practitioners as well as patients to scan their x-ray for possible brain tumours. We hope to learn about the various machine learning algorithms that aid the process of image classification and evaluate the best possible algorithm for the task at hand, specifically the CNN and Transfer Learning methods.

1.3. Scope

With all the problems discussed above, the app will find itself as a core part of the health industry. There are many areas of the app that can be fine tuned. For example, adding a personal profile where changes in the tumour structure of a particular individual in a series of brain scans can better analyse the progression of the tumour. Another example would be telemedicine, i. e. the remote diagnosis and treatment of patients by means of telecommunications technology. In rural areas where it's not possible to consult good oncologists given the lack of well-equipped hospitals, we can introduce scanning equipment and an ML model trained to detect tumours hence eliminating a doctor's time spent analysing scans and reports and reducing human error. These advancements can lead to improved patient care, better clinical outcomes, and more efficient and effective brain tumour management.

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 analyse. Several Clustering and Classification algorithms are aimed at enhancing the prediction accuracy of the diagnosis process in detecting abnormalities. The summaries of each of the papers that talk about aforementioned algorithms are provided below.

B. Sathya and R. Manavalan, Image Segmentation by Clustering Methods: Performance Analysis, International Journal of Computer Applications (0975 – 8887) Volume 29– No.11, September 2011. [3]

Sathya et al. (2011) provided a different clustering algorithm such as K-means, Improvised K-means, C-means, and improvised C-means algorithms. Their paper presented an experimental analysis for massive datasets consisting of unique photographs. They analysed the discovered consequences using numerous parametric tests.

Devkota, B. & Alsadoon, Abeer & Prasad, P.W.C. & Singh, A.K. & Elchouemi, A. (2018). Image Segmentation for Early Stage Brain Tumour Detection using Mathematical Morphological Reconstruction. Procedia Computer Science. 125. 115-123. 10.1016/j.procs.2017.12.017. [4]

B. Devkota et al. have proposed that a computer-aided detection (CAD) approach is used to spot abnormal tissues via Morphological operations. Amongst all different segmentation approaches existing, the morphological opening and closing operations are preferred since it takes less processing time with the utmost efficiency in withdrawing tumour areas with the least faults.

Kaur, Jaskirat & Agrawal, Sunil & Renu, Vig. (2012). A Comparative Analysis of Thresholding and Edge Detection Segmentation Techniques. International Journal of Computer Applications.vol. 39.pp. 29-34. 10.5120/4898-7432. [5]

Jaskirat Kaur et al. (2012) defined a few clustering procedures for the segmentation process and executed an assessment on distinctive styles for those techniques. Kaur represented a scheme to measure selected clustering techniques based on their steadiness in exceptional tenders. They also defined the diverse performance metric tests, such as sensitivity, specificity, and accuracy.

Krizhevsky, A., Sutskever, I., Hinton, G.E., et al. (2012) ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems, 141, 1097-1105. [6]

Krizhevsky et al. 2012 achieved state-of-the-art results in image classification based on transfer learning solutions upon training a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. The neural network, which had 60 million parameters and 650,000 neurons, consisted of five convolutional layers, some of which were followed by max-pooling layers, and three fully-connected layers with a final 1000-way Softmax.

Pan, S. and Yang, Q. (2010) A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22, 1345-1359. [7]

Pan & Yang 2010 survey focused on categorising and reviewing the current progress on transfer learning for classification, regression and clustering problems. In this survey, they discussed the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multitask learning and sample selection bias, as well as covariate shift. They also explored some potential future issues in transfer learning research.

CHAPTER 3

Technology Stack

Python is one of the most popular programming languages for machine learning projects due to several reasons; with the primary one being ease of use and a rich ecosystem of libraries and tools for scientific computing, data manipulation, and visualisation, such as NumPy, Pandas, and Matplotlib. Python is known for its simple and readable syntax, making it easy to understand and write code. It has a large number of libraries and frameworks specifically designed for machine learning, such as TensorFlow, Keras, PyTorch, and scikit-learn, which provide high-level APIs for building machine learning models and handling data.

The libraries we've used for our project are:-

1. Pillow for image processing
2. TensorFlow for data automation, model tracking, and performance monitoring [8]
3. NumPy for handling large multi-dimensional arrays and functions to operate on them
4. pandas for data manipulation and analysis
5. Streamlit to rapidly build and deploy machine learning web apps

Our process of building a machine learning model can be broken down into the following 8 steps: collection and preparation, choosing an ML algorithm, training the model, validating the model, testing the model

3.1. Collection and Preparation of Data

The first step is to obtain a labelled dataset of brain scan images, where each image is labelled as either "tumour" or "non-tumor." This dataset will be used for training, validating, and testing the machine learning model. We have used an open-source dataset from Kaggle that comes with labelled images of brain scans. We then processed the data by resizing the images, normalising pixel values, and splitting it into training, validation, and test sets.

3.2. Choosing a Machine Learning Algorithm

Commonly used algorithms for image classification tasks include convolutional neural networks (CNNs), which are designed to process visual data, and support vector machines (SVMs), which can be used for binary classification tasks. We've used CNN which are designed to automatically learn features from input data, particularly images, by leveraging convolutional and pooling layers to capture spatial hierarchies and reduce the need for handcrafted features.

3.3. Training the Model

This process involves feeding the labelled training images into the model and iteratively adjusting the model's parameters to minimise the prediction error. We tried experimenting with different hyperparameters to optimise the model's performance.

1. The learning rate determines the step size at which the model updates its parameters during training. A higher learning rate may result in faster convergence but could also risk overshooting the optimal parameter values, while a lower learning rate may result in slower convergence or getting stuck in local optima.
2. The size of the pooling windows in the pooling layers of the CNN also plays a pivotal role in the fitting of the model. Larger pooling windows may result in more aggressive down-sampling and reduced spatial resolution, while smaller pooling windows may preserve more details but could increase the risk of overfitting.
3. The number of epochs, or iterations over the entire dataset during training, is also a hyperparameter. Too few epochs may result in underfitting, while too many epochs may result in overfitting.

3.4. Validating the model

Once the model was trained, we evaluated its performance on the validation dataset. This helped us assess the model's generalisation capability and identify any overfitting issues. This allowed us to adjust the model's hyperparameters as needed to improve its performance.

3.5. Testing the model

After validating the model, we evaluated its performance on the test dataset, which provided an independent assessment of the model's accuracy. This helps in estimating the model's real-world performance and in making decisions about its deployment.

3.6. Deploying the model

Once we had a model, we were able to save it and deploy it on a web app using the Streamlit framework. The deployment of the web app is heavily based on libraries as the primary focus of our project was to learn about the machine learning algorithms used in classifying images, especially in the medical sector to aid early detection of brain tumour.

3.7. Proposed Workflow

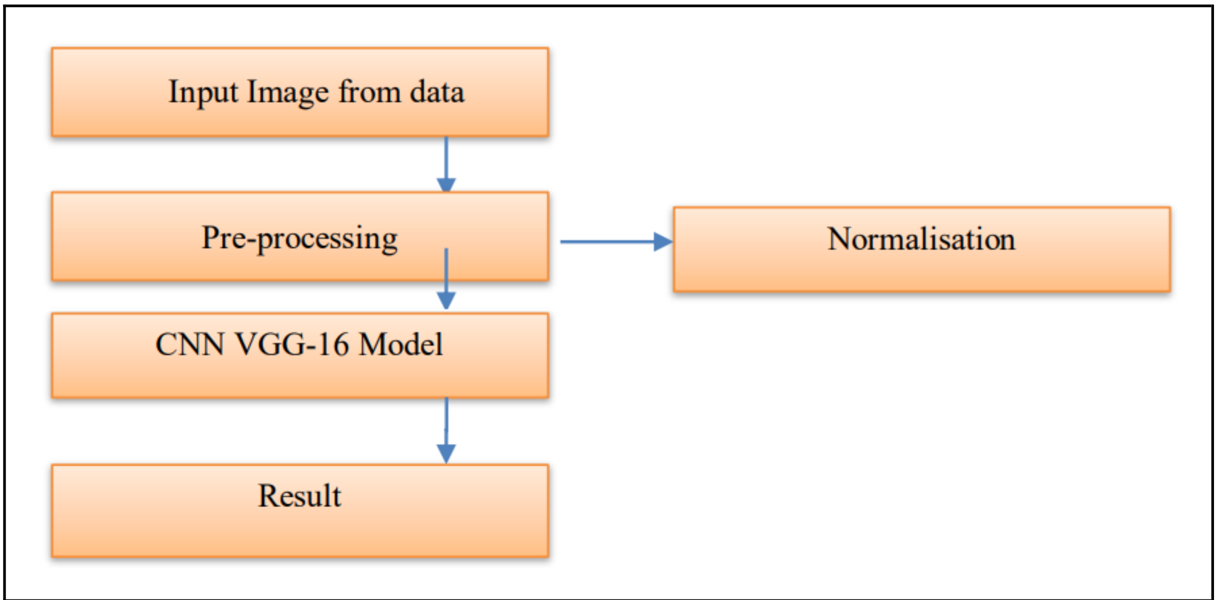

Figure 3.7.1: Proposed Workflow

CHAPTER 4

Benefits and Applications

4.1. Benefits For Society

Firstly, early detection of brain tumours through the use of a reliable and accurate app can lead to early medical intervention, which can potentially save lives. Brain tumours can be life-threatening if not diagnosed and treated promptly. With timely detection, patients can receive appropriate medical care, including surgery, radiation therapy, or chemotherapy, which can improve their chances of successful treatment and recovery. Secondly, it can also help reduce healthcare costs by enabling more efficient and targeted diagnostic processes. Furthermore, a brain tumour detection app can facilitate remote and accessible healthcare, particularly in underserved or remote areas where access to specialised medical expertise may be limited.

4.2. Benefits For Environment

Although the app has no direct influence on the environment, it can indirectly benefit the environment by reducing the need for unnecessary medical tests and procedures, thereby potentially lowering the environmental impact associated with healthcare activities. In addition, a brain tumour detection app that enables remote and accessible healthcare can potentially reduce the need for patients to travel to healthcare facilities resulting in reduced carbon emissions associated with transportation, as well as other environmental benefits.

4.3. Applications

Our web app can easily transform to have a very specific purpose. For example, a personal profile where changes in the tumour structure of a particular individual in a series of brain scans can better analyse the progression of the tumour. Another example would be telemedicine, i. e. the remote diagnosis and treatment of patients by means of telecommunications technology. In rural areas where it's not possible to consult good oncologists given the lack of well-equipped hospitals, we can introduce scanning equipment and an ML model trained to detect tumours hence eliminating a doctor's time spent analysing scans and reports and reducing human error. These advancements can lead to improved patient care, better clinical outcomes, and more efficient and effective brain tumour management.

CHAPTER 5

Project Implementation

5.1. Creating the model

Importing the libraries:-

```
import numpy as np
import cv2
from PIL import Image
import os
from sklearn.model_selection import train_test_split
```

Figure 5.1.1: Libraries

A user defined function “appendImg” to import tumour and non-tumour images from the OS’s file system. The function takes 3 parameters (url, path, target) where ‘url’ is a list of the image names in the directory specified by ‘path’ and the target parameter decides whether the given path and url point to pictures of tumour or no tumour.

```
def appendImg(url,path,target):
    images=[]
    labels=[]
    for i in range(len(url)):
        image=path+url[i];

        img=cv2.imread(image)
        img=img/255
        if img is None:
            print()
        else:
            image=cv2.resize(img,(200,200))

        images.append(image)
        labels.append(target)
    images=np.asarray(images);
    labels=np.asarray(labels);

    return images,labels
```

Figure 5.1.2: A function to add images and labels into a python list for processing

Using the above function, we import the images and their target into a list and stack the results vertically using NumPy's r_() function. At the end, we use the aggregated data to split our dataset into training and testing in the ratio 80:20.

```
# images with tumour
tumourimage, ttarget=appendImg(turl,tpath,1)
# images without a tumour
normalimage, ntarget=appendImg(Nurl,Npath,0)
data=np.r_[tumourimage,normalimage]
targets=np.r_[ttarget,ntarget]
```

Figurer 5.1.3: Stacking classified dataset and targets into respective arrays

Creating a model with multiple layers to compile and train the model and saving it for later use.

```
model = Sequential([Conv2D(32,3,input_shape=(200,200,3),
activation='relu'),
                    MaxPooling2D(),
                    Conv2D(16,3,input_shape=(200,200,3),
activation='relu'),
                    MaxPooling2D(),
                    Conv2D(16,3,input_shape=(200,200,3),
activation='relu'),
                    MaxPooling2D(),
                    Flatten(),
                    Dense(1024, activation='relu'),
                    Dense(512,activation='relu'),
                    Dense(1,activation='sigmoid')
                   ])
model.compile(optimizer='adam',loss=tf.keras.losses.BinaryCrossentropy(),me
trics=['accuracy'])
model.fit(x_train,y_train,batch_size=16,epochs=4,validation_data=(x_test,y_
test))
model.save("tumour.h5")
```

Figure 5.1.4: Training and saving the model

5.2. Performance Evaluation

On experimentation, it was observed that the proposed methodology seems to be outperformed when compared to all different sets of images. Among all the images, the proposed Convolutional Neural Network (CNN) based approach seems too much better in terms of quality of the output in 128 *128 images when compared to its other sized images which are represented in tables and charts. Table 5.2.1 Represents the true positive, true negative, false positive and false negative values of the proposed approach for different sets of images.

Table 5.2.1: Confusion matrix for different resolution of images to be processed				
Resolution	True Positive	True Negative	False Positive	False Negative
128 x128	83.7%	84.5%	16.3%	15.5%
256 x 256	82.4%	84.1%	17.6%	15.9%
512 x 512	82.1%	83.9%	17.9%	16.3%

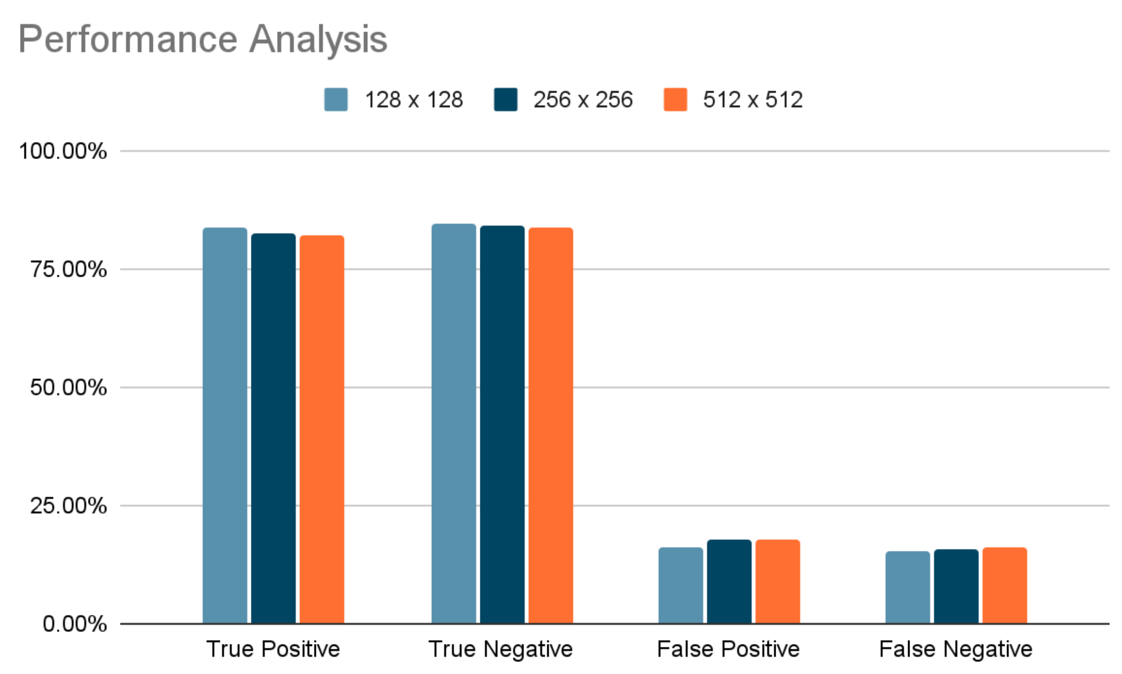

Figure 5.2.1: Performance Analysis

CHAPTER 6

Results

The result of our project is a web app deployed using Streamlit that is able to take an image as an input and identify whether the entered brain scan contains a tumour. Following are the screenshots of our working project:-

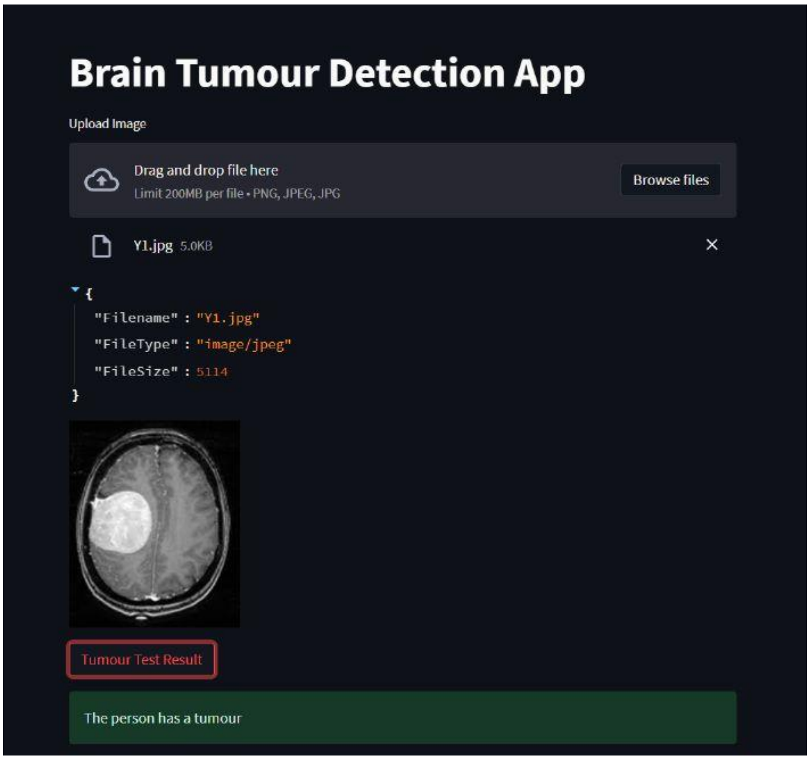

Figure 6.1: User interface

Validation loss refers to the error or loss calculated on a separate validation dataset which is used to evaluate the model's performance on unseen data and to monitor the model's ability to generalise to new data. A lower validation loss indicates better performance, as it means the model's predictions are closer to the ground truth. Validation accuracy is a measure of the model's accuracy or correctness in predicting the target output, calculated as the percentage of correctly predicted samples out of the total number of samples in the validation dataset. Here is the analysis of our model's validation:-

```
... Epoch 1/4
    155/155 [==============================] - 56s 334ms/step - loss: 0.4317 - accuracy: 0.8048 - val_loss: 0.2320 - val_accuracy: 0.9045
    Epoch 2/4
    155/155 [==============================] - 52s 333ms/step - loss: 0.1851 - accuracy: 0.9291 - val_loss: 0.0970 - val_accuracy: 0.9725
    Epoch 3/4
    155/155 [==============================] - 51s 326ms/step - loss: 0.0759 - accuracy: 0.9725 - val_loss: 0.0758 - val_accuracy: 0.9644
    Epoch 4/4
    155/155 [==============================] - 50s 324ms/step - loss: 0.0292 - accuracy: 0.9899 - val_loss: 0.3559 - val_accuracy: 0.8997
```

Figure 6.2: Fitting the model with loss and accuracy analysis

CHAPTER 7

Conclusion

Conclusion

Without a pre-trained CNN model, the training accuracy is 97.5% and validation accuracy is 90.0%. The validation result had a best figure of 91.09% as accuracy. We have also used the VGG-16 CNN model, the training accuracy is 82.2% and validation accuracy is 79.3%. The validation result had a best figure of 80.5% as accuracy. CNN gives better accuracy, boosts the system's performance and thus, is better than transfer learning (VGG-16) when it comes to image detection. In summary, we propose A CNN based method for segmentation of brain tumours in MRI images. There are a number of techniques available for brain tumour segmentation and classification to detect the brain tumour. To overcome these limitations, proposed is a Convolution Neural Network (CNN) based classifier. The CNN based classifiers are used to compare the trained data and test data, from this data we get the best result.

References

[1] “Cancer Statistics.” *National Cancer Institute*, https://www.cancer.gov/about-cancer/understanding/statistics.

[2] Feb 5, 2021. “More than 50% Cancer Deaths Due to Late Diagnosis, Say Docs: Madurai News - Times of India.” *The Times of India*, TOI, https://timesofindia.indiatimes.com/city/madurai/more-than-50-cancer-deaths-due-to-late-diagnosis-say-docs/articleshow/80697048.cms.

[3] Sathya, Bharath, and R. Manavalan. "Image segmentation by clustering methods: performance analysis." *International Journal of Computer Applications* 29.11 (2011): 27-32.

[4] Devkota, B., et al. "Image segmentation for early stage brain tumor detection using mathematical morphological reconstruction." *Procedia Computer Science* 125 (2018): 115-123.

[5] Kaur, Jaskirat, Sunil Agrawal, and Renu Vig. "A comparative analysis of thresholding and edge detection segmentation techniques." *International journal of computer applications* 39.15 (2012): 29-34.

[6] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Communications of the ACM* 60.6 (2017): 84-90.

[7] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2010): 1345-1359.

[8] “Introduction to Tensorflow.” *TensorFlow*, https://www.tensorflow.org/learn.