**1. Introduction**

**1.1 Introduction**

* This document lays out a project plan for accomplishing “Brain tumour detection and identification using Machine Learning”, by Hirva Bhagat and Rutvi Patel
* This document is stating all the necessary information needed to understand the meaning behind the above-mentioned project and our ideology for achieving it.
* Our focus is to create a software that could help in detecting brain tumour by taking Brain MRI as input. It would use machine learning functions to segment the brain tumour from the image and identify it.

**1.2 Aim and Objectives of the work**

* Our aim is to create a software that could accurately segment and identify brain tumour from MRI images. This will work to help to ease the burden of manually going through MRI and documenting it.

*Objectives***:**

* + - Label data as yes and no depending on presence of tumour
    - Augment data the labelled data in proportion to angle, light and crop to make training easier
    - Training the model using the augmented labelled MRI dataset. We could use GPU here to achieve greater training speed.
    - Validate the trained model to determine accuracy
    - Testing the dataset by giving a test MRI image as input
    - Verifying the result and calculating error % to measure accuracy
    - Integrating model with an interface to make it easier to use

**1.3 Scope**

* users should be able to train their own model by providing labelled training data
* users should be able to use the pretrained model if they desire
* users should be able to input validation dataset or an image to verify if tumour is present or not
* user should be able to see the output and result specifying the detection of tumour if present or they should get a message which conveys that no tumour was found

**1.4 Stakeholders**

The segmentation, detection, and extraction of infected tumour area from magnetic resonance (MR) images are a primary concern but a tedious and time-consuming task performed by radiologists or clinical experts, and their accuracy depends on their experience.

* Radiologists: Radiologists can use this system to easily detect brain tumours. Further helping them to take necessary treatment in an easy manner with fast decision making. The system is meaningful in deciding the treatments like Chemotherapy and Radiations for the patients and for reducing the workload of doctors by giving accurate results.
* Surgeons: the system can be automatic or semi-automatic for brain tumour segmentation. Also, as a pre requisite stage for doctors to identify the brain tumour before performing surgeries
* Doctors and Patients: Tumour grading after detection can help doctor and patient or family members to understand the patient’s condition. Also helping doctors to plan treatment and predict outcome

Detection is the initial stage; the progression of this stage is followed by segmentation which can help doctors grasp the exact progression of the disease state and make appropriate decision and improve diagnostic accuracy of para-medical analysis

**1.5 Brief Literature Review**

Brain tumour is a benign or malignant growth of abnormal cells in brain. Primary brain tumours originate from cells within the brain and secondary (metastatic) brain tumours begin in another part of the body and then spread to the brain. As MRI provides high resolution and high contrast images of the brain in the axial, coronal and sagittal orientation providing a three-dimensional assessment of the lesion and greater contrast between different soft tissues of human body, it is preferred over computed tomography (CT) and other techniques. It is found to be more effective in examining tumour images and gives accurate results.

**Literature Review:**

Method proposed by [1] begins by converting the image to grayscale and removing noise from it. To enhance the image, they perform edge detection and add it to the original image. From the enhanced image the tumour is segmented from the background using thresholding. Lastly, Morphological operations and windowing technique are used to locate the segmented part in the image. Testing dataset is made up of 100 MR images and segmentation accuracy achieved is 97%.

The method proposed by [2] identifies brain tumour of three kinds: glioma, meningioma and pituitary tumour using MRI images. The dataset used is stored in MATLAB of size 350× 350 in JPEG image format and displayed as RGB gray scale image where the entries ranging from 0 to 1. Here, the denoised RGB gray scale images are converted to HSV format to process the images. The first stage, i.e. pre-processing contains three sub process: Image enhancement, Filter operation and segmentation. Image enhancement uses contrast stretching and it gives better results on gray scale images so as contrast is increased without distorting relative gray level intensities. This eliminates the ambiguity appearing in different regions of the image. In filter operation, Median Filter is applied, which removes the noise at satisfactory level while preserving the edges of the image. Then, Otsu thresholding method is used for segmentation.

Post-processing activities include feature extraction and identification. Area, circularity and solidity features were considered as the main examining features and PCA feature-selection algorithm was applied. The main aim of identification process is to detect the ROI by using the features extracted in the feature extraction section. Some sizes are predefined for the feature measurement and a range of values of sizes is set for the features and when a region falls within the value range of each feature then, the ROI is identified successfully. The study claims that the accuracy rate of proposed method is comparatively more than the other existing approaches.

[3] introduces Anisotropic diffusion as a preprocessing step, which overcomes the loss in image detail and features caused due to denoising. Skull stripping or exclusion of grey and white matter from MR images is done by using Otsu's thresholding. Segmentation of tumour is achieved by using Fuzzy c-means clustering modified by multiplicative low-frequency bias field estimation.

**1.6 Problem definition**

The main purpose of this analysis is to demonstrate the concept and documentation of software which identifies and segments brain tumour from MRI images using machine learning. Also, to study implementation of machine learning on image data and understand the struggles faced in data preparation as well as what factors weigh into determining the accuracy of the model.

**1.7 Plan of work**

* Dataset: link: [ [https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumour-detection](https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection) ] The dataset contains 2 folders: yes and no. The folder yes contains 155 Brain MRI Images that are tumourous and the folder no contains 98 Brain MRI Images that are non-tumourous. In total the dataset contains 253 Brain MRI Images.
* Augmentation:

Since the amount of data is very less to train, we use augmentation to increase the size of the dataset and also tackle the data imbalance issue.

1. For this we use keras (keras.preprocessing.image ImageDataGenerator).
2. Before data augmentation:

155 positive and 98 negative examples, resulting in 253 MRI images.

1. After data augmentation:

1085 positive and 980 negative examples, resulting in 2065 MRI images.

* Data Preprocessing:
  + Cropping the MRI so that it contains only the brain (most important part of the image).
  + Resize the MRI to have a shape of (240, 240, 3)=(image\_width, image\_height, number of channels). So all images have similar shape.
  + Apply normalization: to scale pixel values to the range 0-1. To make it easier to detect tumour.
* Splitting the data:
  + 70% of the data for training.
  + 15% of the data for validation.
  + 15% of the data for testing.
* Training: using CNN in python (Tenserflow and Keras)( The model was trained for 24 epochs and these are the loss & accuracy plots)

Architecture:

* + A Zero Padding layer: pool size of (2, 2).
  + A convolutional layer: with 32 filters, a filter size of (7, 7) and a stride equal to 1.
  + A batch normalization layer.
  + A ReLU activation layer.
  + 2 Max Pooling layers with f=4 and s=4.
  + A flatten layer: (3D matrix into a 1D vector).
  + A Dense (output unit) fully connected layer with one neuron with a sigmoid activation.
* Segmentation: K-means or Fuzzy C. If implementation using this is not possible then UNet will be used for implementation.
* Interface: python(Tkinter)

**2. TECHNOLOGY AND LITERATURE REVIEW**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr No.** | **Problem Description** | **Segmentation technique** | **MRI** | **Segmented Image** | **Result** |
| 1 | Brain tumour segmentation based on a hybrid clustering technique | K-means clustering technique integrated with Fuzzy C-means algorithm, followed by thresholding and level set segmentation stages |  |  | This technique provides two benefits-using K means clustering minimize computation time and Fuzzy C means provides better accuracy. |
| 2 | Automated Brain Tumour Segmentation and Detection in MRI using Enhanced Darwinian Particle Swarm Optimization (EDPSO) | Enhanced Darwinian Particle Swarm Optimization (EDPSO) |  |  | Comparative analysis is based on execution time and accuracy of proposed EDPSO algorithm with PSO Algorithm. From the results obtained EDPSO Algorithm received a better quantity rate for all the input images. |
| 3 | An Integrated Design of Particle Swarm Optimization (PSO) with Fusion of Features for Detection of Brain Tumour | Particle swarm optimization (PSO) |  |  | LBP and deep features are extracted from the image using PSO. |
| 4 | Performance Evaluation of Fuzzy C Means Segmentation and Support Vector Machine Classification for MRI Brain Tumour | Fuzzy C-means algorithm |  |  | 98.2% accuracy is obtained. |
| 5 | Brain tumour detection based on Convolutional Neural Network with  neutrosophic expert maximum fuzzy sure entropy | Neutrosophic set – expert maximum fuzzy-sure entropy  (NS-EMFSE) approach |  |  | NS-EMFSE provides better accuracy, sensitivity, specification and Youden index of the MRIs in comparison to other segmentation techniques to the classifiers. |
| 6 | A Fuzzy clustering based MRI brain image segmentation using back  propagation neural networks | Spatial fuzzy clustering  algorithm |  |  | The process is intended to segment the tumour cells as well as the  removal of background noises for smoothening the region, results in presenting segmented tissues and parameter evaluation  to produce the algorithm efficiency. |
| 7 | Computer Aided System for Brain Tumour  Detection and Segmentation | Global threshold  segmentation |  |  | Segmentation accuracy is 97% |
| 8 | Fuzzy Clustering and Deformable Model for Tumour Segmentation on MRI  Brain Image: A Combined Approach | Enhanced  Possibilistic Fuzzy C-Means (EPFCM) and one of the contour-based methods called parametric deformable model with gradient  vector field (GVF) as an external force field |  |  | It is quantitatively verified that the combined approach provides better segmentation result. |
| 9 | MRI brain lesion image detection based on colour-converted K-means  clustering segmentation | Colour-converted segmentation algorithm with K-means clustering technique |  |  | This method gives encouraging results which allows  pathologists to distinguish exactly lesion  size and region as the  brain regions related to a tumour or lesion can be exactly  separated from the colored image. |
| 10 | Brain Tumour Segmentation and Its Area  Calculation in Brain MR Images using K-Mean  Clustering and Fuzzy C-Mean Algorithm | Advanced K-means  and Fuzzy C-means algorithms |  |  | Tumour is extracted using K-means clustering and Fuzzy C means is used for accurate  tumour shape extraction of malignant tumour. |

**3. SYSTEM REQUIREMENTS STUDY**

**3.1 User Characteristics**

* It makes the job of medical professionals easy as tumour can be detected easily with an accuracy rate 88% at an early stage if diagnosed at right time.
* Even a layman can use their MRI to identify tumour on their own.
* It can be automated to do detection in real time and further integrated with high end hardware to perform identification in bulk

**3.2 Hardware and Software Requirements**

* Hardware:

1. GPU (to train from google Collab)

* Software:

1. Python (libraries-especially Keras & Tensorflow for machine learning)
   1. **Assumptions and Dependencies**

* **Assumptions:**

1. **Technology:** we assume that the system supports python and has all libraries and environment needed to run the software
2. **Machine Learning model:**

* **Independent and identically distributed data (IID) data**: This is a common assumption in supervised and unsupervised learning. Essentially assuming that there is some true but unknown data distribution from which each of your training and test points are drawn independently.
* **The true function mapping inputs to outputs is smooth almost everywhere**: Essentially, you assume that there is some true but unknown function that maps the inputs to the outputs
* **Signal to noise ratio is large**: The noise in input features and labels is assumed to be low. Of course, if the data you’re learning from is very noisy, you cannot learn useful models.
* **Unmodelled features have negligible effects**: You represent any input as a set of features and there are other variables in the environment that you ignore in the process, we assume that they have negligible effect on output.

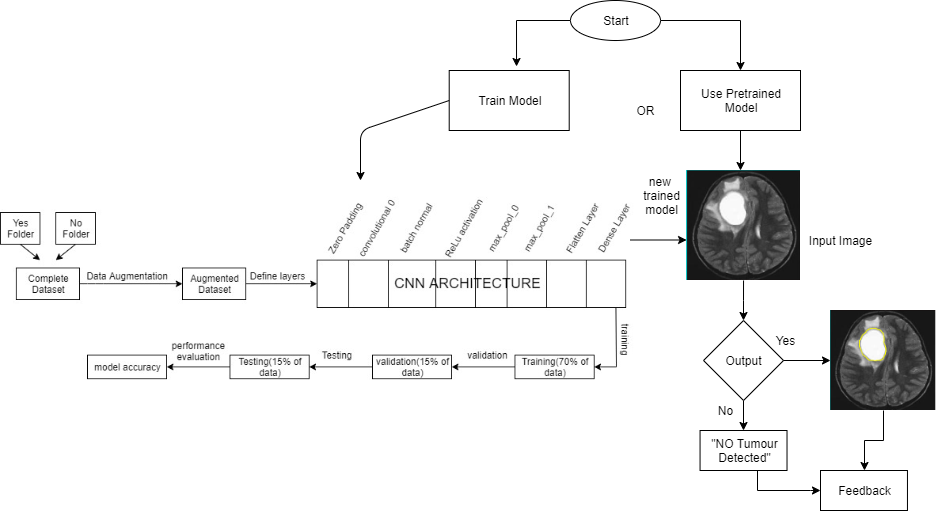
1. **Data:** if not using the pretrained model, we assume that the data provided for training is MRI and are correctly labelled before augmentation.
2. **User:**

* User is assumed to have enough knowledge about giving input and selecting model/ training model
* It is assumed that user will provide an MRI to validate
* **Dependencies:**

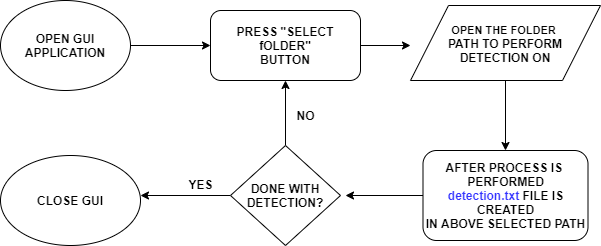
1. System is dependent upon the valid training data
2. System is dependent upon the correct format of the input and will not process otherwise
3. System accuracy depends upon the training data provided

**4. SYSTEM DIAGRAM**

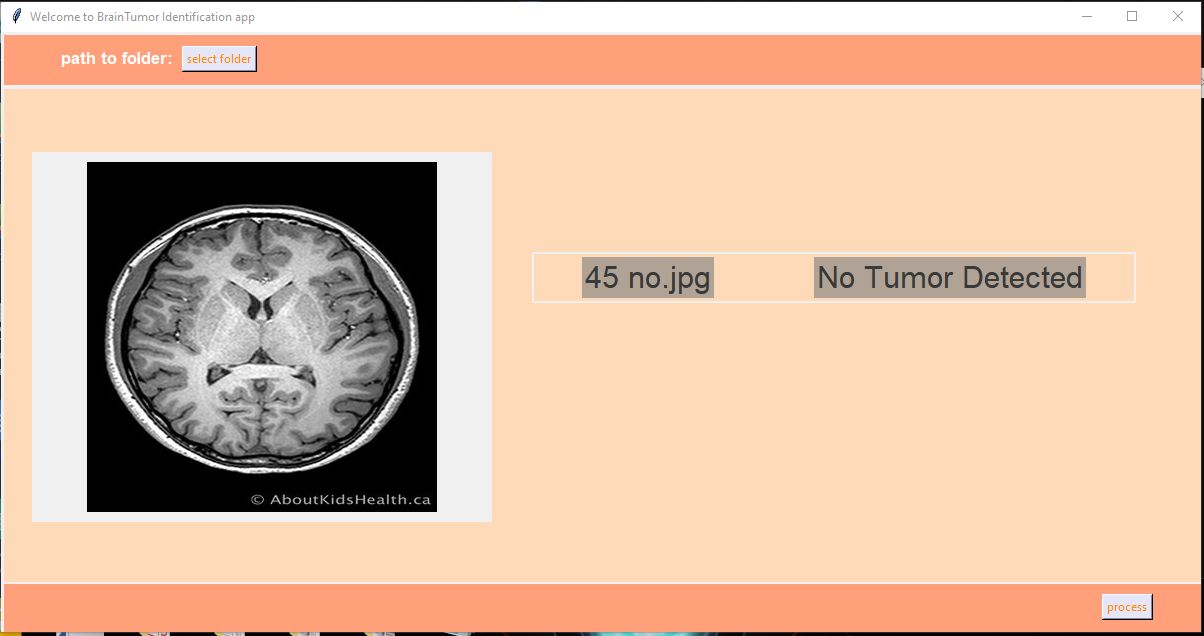
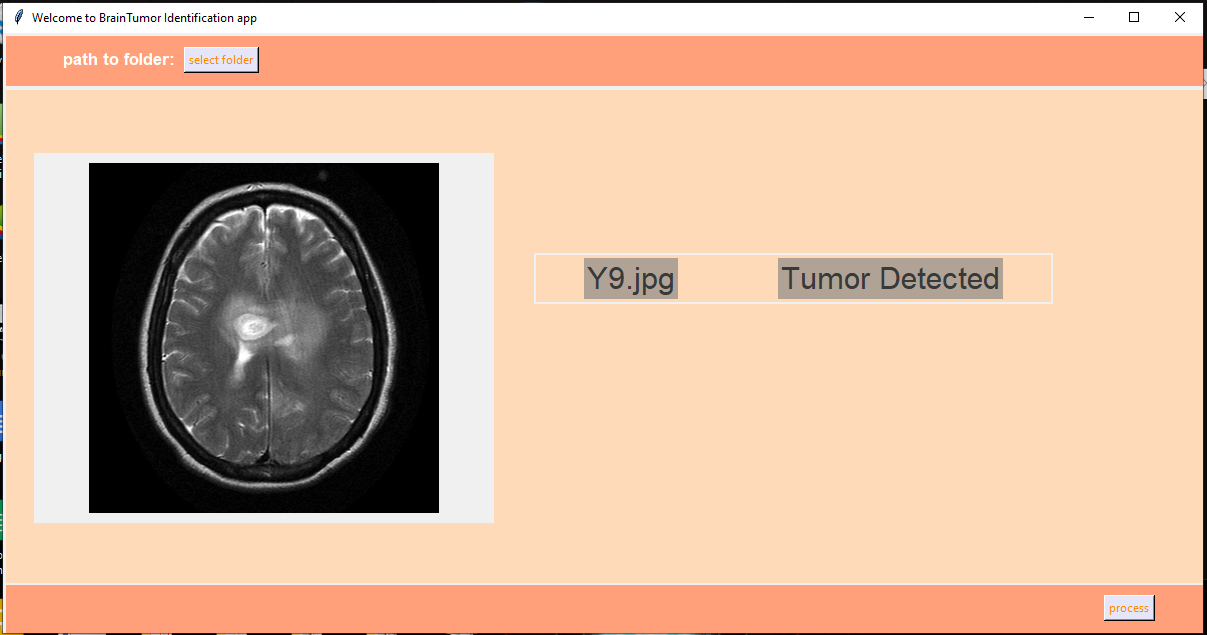
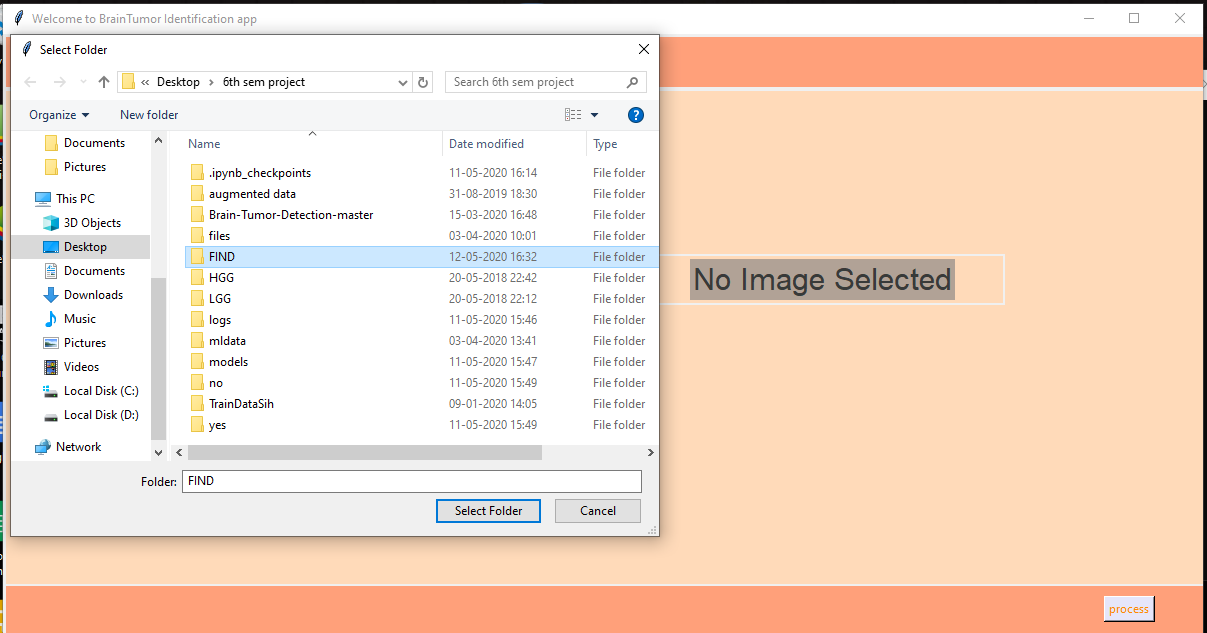
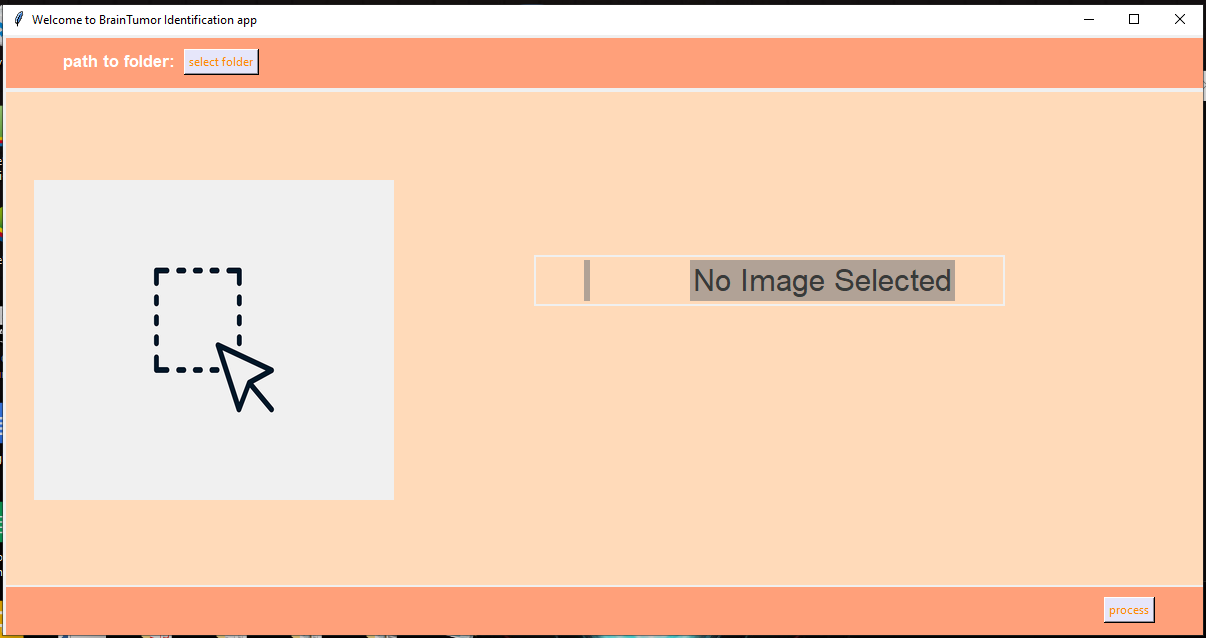
* **System flow diagram**

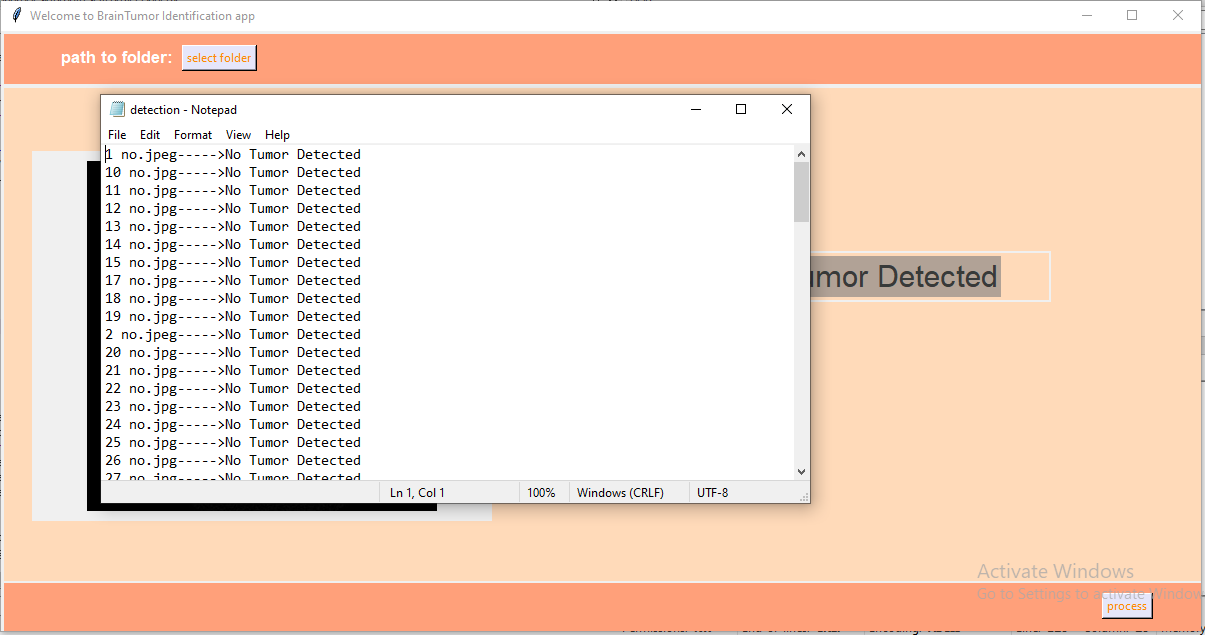
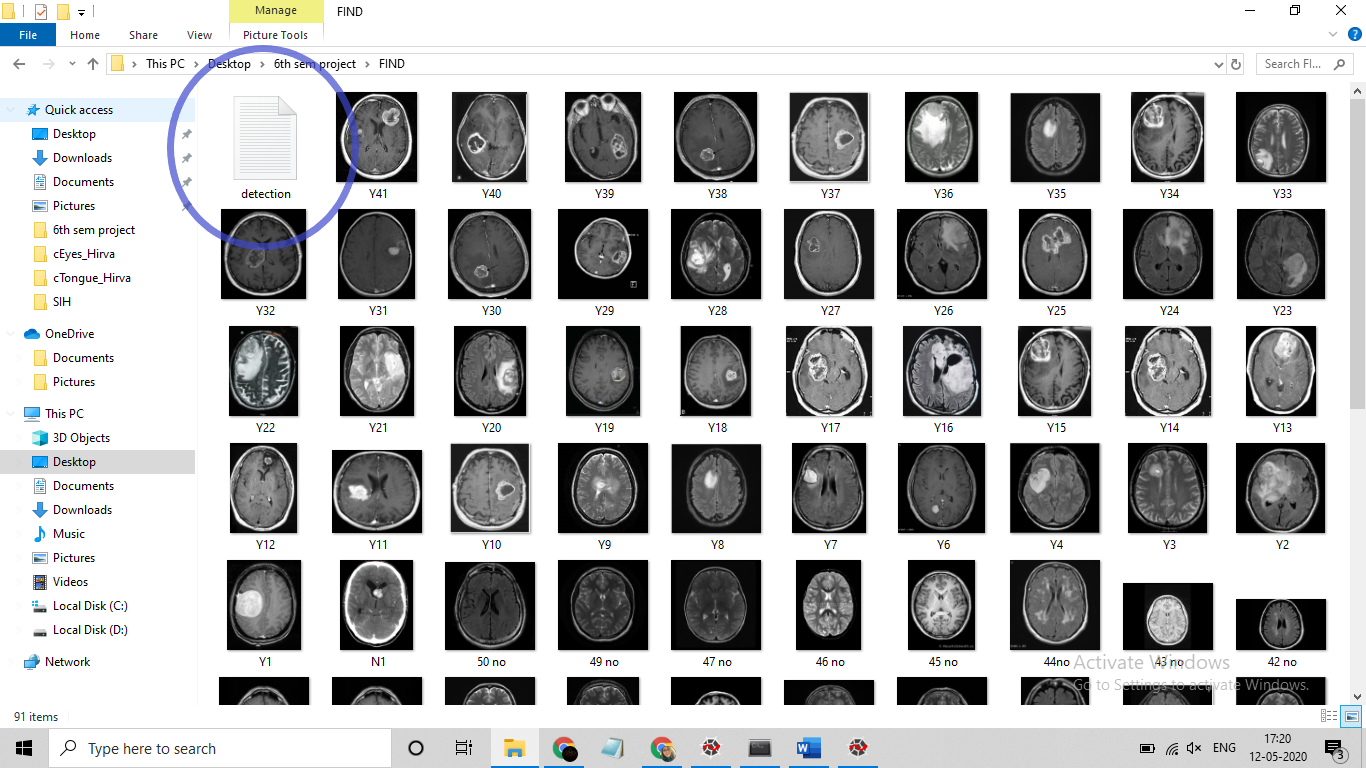
****

* **Gui Flow diagram**

****

**5. GUI Screenshots**

****

****

**6. PERFORMANCE METRICS**

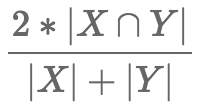
* **EVALUATE**

**evaluate(**x=None, y=None, batch\_size=None, verbose=1, sample\_weight=None, steps=None, callbacks=None, max\_queue\_size=10, workers=1, use\_multiprocessing=False**)**

Our arguments: loss, acc = best\_model.evaluate(x=X\_test, y=y\_test)

**Arguments**

1. x: Input data.
2. y: Target data.
3. batch\_size: Number of samples per gradient update. If unspecified, batch\_size will default to 32.
4. verbose: 0 or 1. Verbosity mode. 0 = silent, 1 = progress bar.
5. sample\_weight: Optional Numpy array of weights for the test samples, used for weighting the loss function.
6. Steps:Total number of steps (batches of samples) before declaring the evaluation round finished. Ignored with the default value of None.
7. Callbacks: List of callbacks to apply during evaluation.
8. max\_queue\_size: Maximum size for the generator queue. If unspecified, default to 10.
9. workers: Maximum number of processes to spin up when using process-based threading. If unspecified, workers will default to 1. If 0, will execute the generator on the main thread.
10. use\_multiprocessing: Used for generator or keras.utils.Sequence input only. If True, use process-based threading. Default false.

Keras uses Dice Coeffecient to evaluate:

The ∩ sign stands for [set intersection 67](https://en.wikipedia.org/wiki/Intersection_(set_theory)), and |X| stands for the [cardinality 127](https://en.wikipedia.org/wiki/Cardinality) of the set X – basically the number of elements in the set.

This coefficient measures the similarity between sets X and Y. If the two sets are identical (i.e. they contain the same elements), the coefficient is equal to 1.0, while if X and Y have no elements in common, it is equal to 0.0. Otherwise it is somewhere in between. The reason intersection is implemented as a multiplication and the cardinality as sum() is because **pred** and **target** are vectors consisting of zeros or ones.

* **F1\_SCORE:**

we use f1\_score from sklearn.metrics to calculate f1 score of our model.

*sklearn.metrics.****f1\_score****(*y\_true*,*y\_pred*,*labels=None*,*pos\_label=1*,*average='binary'*,*sample\_weight=None*,*zero\_division='warn'*)*

**F1 = 2 \* (precision \* recall) / (precision + recall)**

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

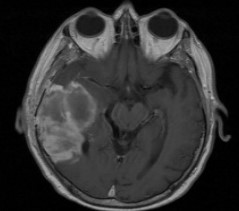
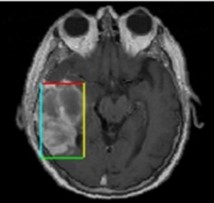
Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the [Positive Predictive Value](http://en.wikipedia.org/wiki/Positive_predictive_value) (PPV).

The [F1 Score](http://en.wikipedia.org/wiki/F1_score) is the 2\*((precision\*recall)/(precision+recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall. where an **F1 score** reaches its best value at 1 and worst at 0.

**7. WEAKNESS AND FEASABILITY**

* 1. **Problems and Weakness in current system**
* **Deployment is difficult:** deploying ML on your own servers or hardware continues to be a challenge and we cannot guarantee that the dependencies of our model is satisfied by the specifications of hardware/OS it will be deployed on.
* **Integration not included:** Integrating prediction API’s into products/workflows is difficult and we have yet to integrate the model into any interface.
* **Not learning:** there is no feedback loop. Machine learning is not “learning” unless there is a continuous feedback loop of data from production
* **Removing Bias:** our model is prone to bias as no feedback is provided.
  1. **Feasibility study**
* **Technical Feasibility:** the technical feasibility centers on existing computer operating system and hardware, and to what extent it can support the proposed model. For example, the versions of python libraries and support used in training the model can vary, depending upon how they are installed in the existing system. This in most cases will not hinder the integration/ smooth running of proposed model but it cannot be ignored that in significant version gaps it is not feasible to run the proposed model in the existing system.
* **Cost feasibility:** the proposed model is cost feasible if the hardware required is present. However, to train the model using huge amount of data will require GPU and that will increase the cost of integrating the system
* **Operational feasibility:** proposed model depends upon labelled dataset to operate. To train the model there is an assumption of availability of labelled data or manpower to label data properly in order to train the model. However, if data is present but not labelled, the proposed model will not be operational as it would not have any valid dataset to train.

**8. RESULT, DISCUSSION AND CONCLUSION**

** **

**(Brain Tumour MRI) (Brain Tumour Detected)**

* **Result:** The training accuracy achieved by our ML model to detect brain tumour from MRI image is 88%
* **Updated Result:** retraining with more epochs (with google collab and GPU). The accuracy achieved is 92.53%
* **Result Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Total images tested** | **Images giving correct result** | **Images giving incorrect result** | **Accuracy=correct/total** |
| **335** | **310** | **25** | **310/335=92.53** |

* **Future Work:** Accuracy can be increased by gathering more qualitative dataset. Deep learning can be further explored to derive a better algorithm for achieving better results
* **Conclusion:**

Working on this project was definitely an informative experience. We faced many difficulties and eventually were introduced with all the struggles faced with training any ML model. We studied all the factors that go into increasing the accuracy and how important it is to preprocess and clean the data. Also, that the availability of high-end hardware specifications and training data is detrimental in making of a good ML model.

**9. REFERENCES:**

[1] M. U. Akram and A. Usman, "Computer aided system for brain tumour detection and segmentation," *International Conference on Computer Networks and Information Technology*, Abbottabad, 2011, pp. 299-302.

[2] T. M. Shahriar Sazzad, K. M. Tanzibul Ahmmed, M. U. Hoque and M. Rahman, "Development of Automated Brain Tumour Identification Using MRI Images," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, 2019, pp. 1-4.

[3] Bilenia A., Sharma D., Raj H., Raman R., Bhattacharya M. (2019) Brain Tumour Segmentation with Skull Stripping and Modified Fuzzy C-Means. In: Satapathy S., Joshi A. (eds) Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 106. Springer, Singapore

[4] Eman Abdel-Maksoud, Mohammed Elmogy and Rashid Al-Awadi, “Brain tumour segmentation based on a hybrid clustering technique”, 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

[5] Vasupradha Vijay, Dr.A.R.Kavitha and S.Roselene Rebecca, “Automated Brain Tumour Segmentation and Detection in MRI using Enhanced Darwinian Particle Swarm Optimization (EDPSO)”, 2016, Elsevier.

[6] Muhammad Sharif, Javaria Amin, Mudassar Raza, Mussarat Yasmin and Suresh Chandra Satapathy, “An Integrated Design of Particle Swarm Optimization (PSO) with Fusion of Features for Detection of Brain Tumour”, 2019 Published by Elsevier B.V.

[7] B. Srinivas and G. Sasibhushana Rao, “Performance Evaluation of Fuzzy C Means Segmentation and Support Vector Machine Classification for MRI Brain Tumour”, *Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing, 2019.

[8] Fatih Özyurt, Eser Sert, Engin Avci and Esin Dogantekin, “Brain tumour detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy”, 2019, Elsevier.

[9] C. Senthilkumar and R. K. Gnanamurthy, “A Fuzzy clustering based MRI brain image segmentation using back propagation neural networks”, Springer Science+Business Media, LLC, part of Springer Nature 2018.

[10] M. Usman Akram and Anam Usman, “Computer Aided System for Brain Tumour

Detection and Segmentation”, 2011, IEEE.

[11] A.Rajendran and R. Dhanasekaran, “Fuzzy Clustering and Deformable Model for Tumour Segmentation on MRI Brain Image: A Combined Approach”, 2012, Elsevier.

[12] Li-Hong Juang and Ming-Ni Wu, “MRI brain lesion image detection based on color-converted K-means clustering segmentation”, 2010, Elsevier.

[13] J.selvakumar, A.Lakshmi and T.Arivoli,” Brain Tumour Segmentation and Its Area Calculation in Brain MR Images using K-Mean Clustering and Fuzzy C-Mean Algorithm”, 2012, IEEE.