2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT).
India Habitat Centre, Lodhi Road, New Delhi, India. Sep 23-25, 2022

Brain Tumor Diagnosis and Classification based on AutoML and Traditional Analysis

Dr. Sindhu P Menon
Professor, Dept. of Computer Science & Engineering
Dayananda Sagar University
Bangalore, India
sindhu33in@gmail.com

KondapalliVaishaali
Dept. of Computer Science & Engineering
Dayananda Sagar University
Bangalore, India
vaishaalikondapalli@gmail.com

Sathvik N G
Dept. of Computer Science & Engineering
Dayananda Sagar University
Bangalore, India
sathvik.182@outlook.com

Siva Prakash Anupam Gollapalli
Dept. of Computer Science & Engineering
Dayananda Sagar University
Bangalore, India
anupam.gollapalli@gmail.com

Subhramanya N Sadhwani
Dept. of Computer Science & Engineering
Dayananda Sagar University
Bangalore, India
subhramanya.sadhwani@gmail.com

Vaishnavi A Punagin
Dept. of Computer Science & Engineering
Dayananda Sagar University
Bangalore, India
va.punagin@gmail.com

2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT) | 978-1-6654-9365-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBCONPT57482.2022.9938146

***Abstract*—Analysis of brain tumors is a challenging task. Due to the complexities of the brain structure, the analysis of a brain MRI (Magnetic Resonance Imaging) requires sophisticated tools and vast knowledge in the field of neuroscience. Detecting the presence of Malignant brain tumors can be done by using ML (Machine Learning) techniques. Moreover, recent changes in automation with the advent of tools like AutoML (Automated Machine Learning) have created significant room for research. When training ML models, there are several hyperparameters such as the number of clusters in the case of KNN, or the number of hidden layers in the case of Artificial Neural Networks. Different methods and models are implemented on the brain MRI scan dataset. The images were preprocessed and ran on all the models discussed in the paper individually. The validation accuracies are then compared with the performance results obtained by using AutoML. The comparison of the obtained validation accuracies assists in discerning the most optimal segmentation method along with the corresponding preprocessing technique used. The best model can then be used for detecting the presence of brain tumors in any given MRI scan.**

Keywords— AutoML, VGG, Otsu, RESNET, KNN, SVM

I. INTRODUCTION

A brain tumor materializes when a cluster of abnormal cells forms within the brain. Brain tumors can either be malignant (cancerous) or benign (noncancerous). The origin of a primary brain tumor is in the brain. The majority of the initial brain tumors are innocuous. The secondary brain tumor which is also referred to as a metastatic brain tumor, happens when cancer cells from another organ, such as your lung or breast, travel up to your brain. The pressure inside your skull can increase whenever benign or malignant tumors get bigger. This can lead to brain damage, which can be fatal too. Doctors use tumor grading to explain the growth and spread of a brain tumor. Grade 1 i.e, low grade, Grade 2 i.e, intermediate grade, Grade 3 i.e, high grade, and Grade 4 i.e, high grade are the several types of grades for a brain tumor. We suggest a system that will categorize if a patient has a brain tumor or not, and if a tumor exists then we classify the type of tumor. The classes of tumors detected are: Glioma, Meningioma, and Pituitary tumor. Machine Learning, Deep Learning and Transfer Learning methods are used in this paper to achieve the goals, which can be beneficial to the healthcare industry. Convolutional Neural Networks, VGG16, RESNET50, KNN, SVM, Logistic Regression, and Ensemble models are the models implemented in the research. Multi Otsu, HSV, Otsu, DWT, and Inverse Otsu are the preprocessing techniques implemented. The purpose of experimenting with various preprocessing approaches is to compare the accuracies gained from all of them, as well as to compare them to the accuracies obtained with raw images, by doing so we can easily find out if a particular pre - processing technique is making a positive impact on the models' accuracies or if it has nothing to add to the models' performance. For feature extraction, BRIEF (Binary Robust Independent Elementary Features) feature point descriptors have been used for drawing out notable features from the MRIs with tumors. AutoML also has been implemented to see how efficacious its performance is compared to the other models. Automated machine learning (AutoML) is a cloud-based machine learning model builder that automates the process of Machine Learning for the purpose of image recognition. AutoML is based on Neural Architecture Search(NAS). It automates the selection, composition, and parameterization of machine learning models in particular. The use of automl was motivated by the desire to improve accuracy while reducing training time, which is reduced because automl does feature engineering and selects the most optimal algorithm for the task on its own.

II. RELATED RESEARCH

Bathe et. al. [1] have compared the accuracies obtained from Depthwise Convolutional Neural Network and other models such as Support Vector Machine, K Nearest Neighbour and standard Convolutional Networks. It was concluded that Depthwise CNNs had the best performance. Tahia Tazin et. al. [2] used deep learning models such as a modified version of MobileNetV2, VGG19, and InceptionV3. The modified MobileNetV2 achieved the highest accuracy. S. Grampurohit et. al. [4] compared the VGG16 and CNN segmentation models over 25 epochs. A. Hussain and A. Khunteta [5] performed semantic segmentation of the brain tumor from MRI images using SVM classification with GLCM features. An average accuracy of 93.05% had been obtained. Tessy George and T. Ramakrishnan at [3] performed image segmentation, morphological operations, and feature extraction for pre-processing and used for detecting brain cancer in the MRI.

Modified morphological based Fuzzy C Means algorithm was used to segment the cancer region. M.Anto Bennet et. al. [6] performed DT-CWT on split video frames and used a neural network to identify brain tumors. S.Somasundaram and R.Gobinath [7] made use of DWT and median for pre-processing and Fuzzy C means for segmentation and with an accuracy of 97.5. Tonmoy Hossain et. al. at [8] Performed skull tripping, segmentation using FCM, morphological contouring, feature extraction and achieved 97.87% validation accuracy on proposed CNN. Kurnar et. al. in [9], calculated the area of the tumor using median-filter. Image segmentation was done using K-means. Mr. T. Sathies Kumar et. al. [10] used SVM for brain tumor detection. Mariam Saii and Zaid Kraitem at [11] used SVM to detect the brain tumor. Swapnil R Telrandhe et. al. [14] performed skull masking, segmentation object labelling and feature extraction to detect presence of brain tumor using the SVM Classifier. Ansari et. al. [12] have used histogram normalization for pre-processing and K-Means for the image segmentation. Nandi and Anupurba at [13] used thresholding, Watershed transformation and Morphological operators for image segmentation, and gray scale conversion along with high-pass filtering for pre-processing. Kailash Sinha and G.R Sinha at [15] performed a comparative study of k-means clustering with watershed segmentation algorithm. Later optimized them with a genetic algorithm and the preprocessing used is image de-noising.Y. Sharma and Y.K. Megharajani at [16], performed brain tumor extraction using mathematical morphological reconstruction. H.S. Abdulbaqi et. al. [17] used Matlab to perform preprocessing using gaussian blur, later HMRF-EM algorithm was used to draw out labels. Used HMRF-EM algorithm over k means.

M. Sharma [18] used histogram equalization, Binarization for pre-processing. Hybrid genetic algorithms and neural network fuzzy inference systems have been used for segmentation.

III. Proposed Methodology

The purpose of this paper is to identify the tumorous region(if present) in the MRIs. According to the prediction made by the model, diagnosis of the tumor will be performed by the health care experts, and will be used for treating the cancer patient.

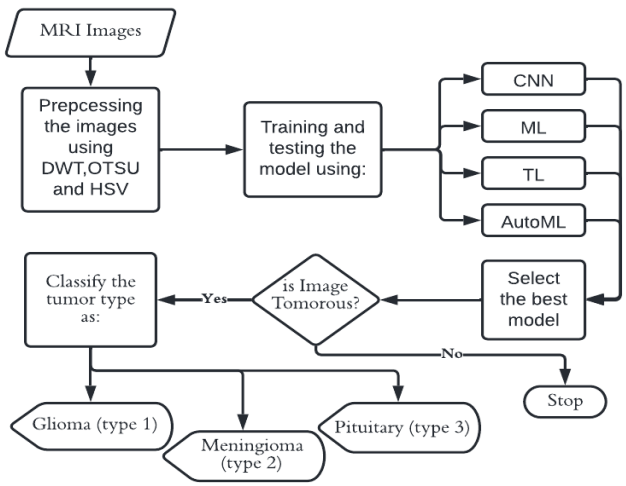

Fig. 1. Flowchart for Brain Tumor Detection and Classification

Fig. 1 shows the flowchart for the work done to present this paper. The preprocessing techniques such as DWT, Otsu, Inverse Otsu, Multi Otsu and HSV are applied to our dataset. All the preprocessed images are run on different algorithms, i.e, KNN, SVM & Logistic Regression, Ensemble model, under Machine Learning(ML),CNN under Deep Learning (DL), and Resnet50 & VGG16 models under Transfer Learning(TL) and Google Cloud AutoML Vision for AutoML.The best model is then chosen manually and considered for evaluation of the results to find out the presence of tumor. Once the presence of tumor is confirmed, it is further classified as to what type of tumor is present i.e Glioma or Meningioma or Pituitary tumor.

A. Splitting the Dataset

The dataset used for training and testing purposes was employed from [19]. The dataset contains MRI data.

For the ML models that have been run, the dataset images used are already split into two parts:

For the ML models,the total number of images used for training is 2583, and the total number of classes is 4, namely, glioma_tumor, meningioma_tumor, no_tumor and pituitary_tumor.

TABLE I. Dataset Split for ML Models

Dataset Division	Tumorous Images			Non–Tumorous Images
	Glioma	*Meningioma*	*Pituitary*	
Training Data	752	735	736	360
Testing Data	74	87	91	35

For the DL and TL models that have been run, the dataset images used are already split into two parts: Training and Testing sets are in the ratio 90:10, wherein under each folder there are four more sub folders containing images of MRI scans of four different types of class, the three tumor types are: Glioma, Meningioma, and Pituitary. The number of images for every class under the Training set and the Testing set are as in Table II:

TABLE II. Dataset Split for TL and DL Models

Dataset Division	Tumorous Images			Non–Tumorous Images
	Glioma	*Meningioma*	*Pituitary*	
Training Data	826	822	827	395
Testing Data	100	115	74	105

The total number of images for the data set used to run the AutoML Model is 1751 as shown in Table III. The split into validation and testing is done automatically depending on the size of the dataset for a particular class by the inbuilt controller that AutoML has.

TABLE III. Dataset Split for AutoML Model

Dataset Division	Tumorous Images			Non–Tumorous Images
	Glioma	*Meningioma*	*Pituitary*	
Training Data	394	380	392	395
Validation Data	49	47	49	33
Testing Data	7	49	49	34

B. Preprocessing

Preprocessing of the input MRIs is done to improve the overall validation accuracies and the efficiencies of these models.

Preprocessing techniques used in this paper include thresholding, color segmentation and wavelet transform.

Otsu: The implementation of Otsu thresholding comprises Binary thresholding along with Otsu thresholding.Implementation for Otsu:

```
res1,Otsu = cv2.threshold(gray,0,255,cv2
.THRESH_BINARY+cv2.THRESH_OTSU)
```

Here, the variable 'gray' refers to the input image converted to grayscale. Other variants of Otsu thresholding are Inverse-Otsu and Multi-Otsu thresholding. MultiOtsu segregates the input image pixels into different classes based on the intensity of gray levels within the image. The "threshold_multiotsu" function from the skimage.filters library is used for the implementation of Multi-Otsu thresholding.

```
Implementation for Multi-Otsu:new_image
=threshold_multiotsu(gray, classes=5)
```

InverseOtsu comprises both Inverse Binary Thresholding with Otsu thresholding.

```
res2,Otsu_inv = cv2.threshold(gray,0,255
,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
```

HSV: HSV color segmentation is used to isolate specific areas of the MRI image of the patient's brain which have similar HSV levels. For the implementation, the albumentations library is used as shown below.

```
transform=albumentations.Compose([
albumentations.HueSaturationValue()])
```

DWT: Discrete Wavelet Transform is a variant of wavelet transform wherein the wavelets are discretely sampled. DWT is essential in de-noising and compressing images. Implementation of DWT on an input image included performing multilevel wavelet decomposition and detail coefficients. Thresholding techniques like sigthresh and wthresh are used in this case. Subsequent to this, the 'imnoise' function available in MATLAB is used to add gaussian white noise to the input image. DWT implementations include Haar wavelets, Daubechies wavelets, and the dual-tree complex wavelet transform. The Haar wavelet, which is a sequence of rescaled "square-shaped" functions form a wavelet family or basis. The dwt2 function computes the multi-level 2-D wavelet decomposition, which is done using the haar wavelet. The following is the model definition of DWT.

```
J = imnoise(input_image, 'gaussian');
[cA1, cH1, cV1, cD1] = dwt2(J, 'haar')
```

The Haar wavelet's mother wavelet function:

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2}, \\ -1 & \frac{1}{2} \le t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Here 'Ψ(t)' is the scaling function and 't' is the time period. t is 1 if it lies between 0 to 0.5, -1 if it lies between 0.5 to 1 and 0 otherwise.

The results of all the preprocessing techniques are discussed in the results section.

C. Feature Extraction using BRIEF

BRIEF (Binary Robust Independent Elementary Features) uses binary strings as descriptors which are vectors portraying the size of features or key points in the MRIs.. The main function of BRIEF is to identify patches surrounding the keypoint and then convert them into a binary vector which represents an object. Therefore, every key point will be described by making use of binary 1's and 0's. The main point to consider about BRIEF is that it is noise sensitive as it deals very closely with the pixel level images. BRIEF easily outperforms other fast descriptors such as SURF and SIFT in terms of speed and terms of recognition rate in many cases. It takes a smoothened patch from the image and then selects a set of nd (x,y) location pairs on which Pixel Intensity comparisons are performed. For example, let first location pairs be p and q. If I(p)<I(q), then its result is 1, else it is 0. This is applied on all the nd location pairs for getting a nd-dimensional bitstring.

D. Models Used

All models that were employed and worked on were coded in python and run on Google Collaboratory.

1) Convolutional Neural Network:

A Six-Layered CNN model has been implemented. Every layer in the model associates with a distinct batch of filters and then aggregates the conclusions before sending it as an output to the next consequent layers in the network. RELU and the sigmoid activation functions were employed to hasten the training period of the neural network.

```
r1=model_cnn.fit(train_x,train_data.label,validation_split=0.1,epochs=20,callbacks=[reduceLR])
```

Here, 'train_x' is what carries the float feature matrix i.e, the design matrix of the training data, and 'train.data_label' is the target/label vector of the training set. The model has been trained for 20 epochs because the validation accuracies remain the same after 20 epochs.

2) Machine Learning Models:

For Logistic Regression, KNN and SVM ML models, linear dimensionality reduction is applied by using the singular value decomposition of the data to project it to a lower dimensional space. To apply PCA on the training set, the n_components value that has been used is 0.98 and 'y_train' is the target or label vector of the training set.

a) Logistic Regression:

The multinomial Logistic Regression is employed to classify MRIs into one of the 4 classes.

The mathematical equation of the general Logistic Regression Classification model is given by

$log[y^{(w,x)} / (1 - y)]$
$= w_0 + w_1 x_1 + \ldots + w_p x_p$

Where $x_1, x_2, \ldots x_p$ are distinct independent predictor values, the distinct features that the Linear Model looks at, and the $w_0, w_1, \ldots w_p$ is designated as w = $\{w_0, w_1, \ldots w_p\}$; a vector which is known as the coefficient and w_0 as the intercept.

```
lg=LogisticRegression()
lg.fit(pca_train,ytrain)
```

b) K - Nearest Neighbors:

In the K-Nearest Neighbour learning method, the label or class for the data point is determined by taking into account a majority of votes on the nearest data point neighbors of each point in the set. The chosen value of k is to indicate the total number of training samples that are needed to classify the test sample. In this case k=3 as shown in the model definition below.

```
knn=KNeighborsClassifier(n_neighbors=3)
        knn.fit(pca_train,ytrain)
```

c) Support Vector Machine Classifier:

With the SVM we aim to generate an optimal hyperplane to separate and classify four classes; the RBF(Radial Bias Function) kernel was used to add extra dimensionality and map the linear data into a higher dimension. The decision points that lie closest to the hyperplane in the passed set of input dataset are called the support vectors, the support vectors positions along with the hyperplane's position are used to classify the test data points. The SVM classifier tries to generate a widest margin to separate classes. Model definition and fit is done as:

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

d) Ensemble model - StackingCV Classifier:

By using stacking, the combination of multiple classification models performances, namely, logistic regression, KNN and the SVM have been used to make this ensemble model.

This classifier makes use of cross validation where the dataset is split into 'k' folds. In the following rounds, 'k-1' folds are used for fitting the first level classifier. In every round, first level classifiers are applied to the left over one subset that has not been used for model fitting in every iteration. Then the final resulting predictions are stacked and given as the input to the second level classifier i.e, the meta classifier, this is what takes in all of the predicted values by lg(logistic regression), knn(k-nearest neighbours), sv(support vector machine) and takes the final call of deciding which class a given data point falls into. The meta classifier used is KNN as it has outperformed the other two machine learning models in terms of overall performance.

```
scv=StackingCVClassifier(classifiers=[
lg,knn,sv],meta_classifier=knn)
scv.fit(xtrain,ytrain)
```

Here, 'xtrain' is what carries the float feature matrix i.e, the design matrix of the training data, and 'ytrain' is the target/label vector of the training set.

3) Transfer Learning Models:

Transfer Learning is a Machine Learning Approach where a model that has been created and trained for one task is now utilized and repurposed as a basis for a different task.

a) ResNet50

ResNet50 is a 50 layered CNN. It is a type of the ResNet model within which 48 layers are convolutional layers and the others are 1 MaxPool layer and 1 AvgPool layer. Once the model is imported, a set split on data is applied to train the model after which the model is run on the test data sequentially.

The weights which have been used for executing the ResNet50 model are from pre - trained imagenet models.

The model definition and fitting it to the training set:

```
base_model = ResNet50(include_top=False, weights = 'imagenet')
```

b) VGG16

A VGG16 model is a 16 layers deep CNN learning model. Amongst the Convolution layers, there also are five MaxPooling layers followed by 3 Fully Connected layers of varied depths, 4096 channels each within the first two, and the third one has 1000-channels to perform the ILSVRC based classification on the given data passed to the model. The last and final layer in this model is a Soft-max layer.

Lesser time and resources are consumed in the application of this model as pretrained weights are used. The model definition and fitting it to the training set:

```
vgg_model= VGG16(weights='imagenet',
include_top=False)
```

Here, 'train_x' is what carries the float feature matrix i.e, the design matrix of the training data, and 'train.data_label' is the target/label vector of the training set. The model has been trained for 20 epochs because the validation accuracies remain the same after 20 epochs.

E. AutoML:

One of our main objectives is to reduce the misclassifications, a robust model like AutoML aids in doing so by resulting in a significantly higher validation accuracy compared to the machine learning and deep learning models implemented. The images are uploaded into Google Cloud Storage Bucket. A new project of type Image classification is created and all the images are imported from Google Cloud Storage Bucket into the newly created project. The images are now manually labelled to different tumor classes that are present and trained on Google Cloud for 20 node hours. The trained model is deployed to a single node with which the model is tested and evaluated.

The evaluation results of all the models are discussed in detail in the next section of this paper.

IV. Results and Discussions

As mentioned, the validation accuracies for each of the ML and TL models are obtained and compared for all the original and pre-processed images. The results are presented in Table IV.

TABLE IV. VALIDATION ACCURACIES FOR DIFFERENT MODELS

Algorithms	No-Preprocessing	OTSU	Inverse Otsu	Multi Otsu	HSV	DWT	DWT-BRIEF
CNN	90.59	12.54	86.06	88.93	87.11	30.51	-
VGG16	90.59	80.14	81.88	87.54	92.68	91.54	-
RESNET50	83.91	36.71	36.94	57.96	58.6	71.79	-
KNN	86.75	85.36	83.62	84.08	79.09	86.39	78.04
SVM	82.57	78.39	79.44	73.01	81.18	87.5	73.17
Logistic Regression	77.7	77.35	75.95	71.62	76.3	86.02	82.92
Ensemble model	87.45	86.06	84.66	74.04	83.27	89.33	85.36

The following are the observations drawn out after comparing the accuracies presented in Table IV:

The DWT pre - processing technique aids all models in performing better and makes a positive impact on their accuracy.Input MRIs for which DWT was applied, resulted in the highest improvement for four out of seven of the models used(excluding AutoML).

The DWT pre- processed images have been fed as input for implementation of BRIEF feature extraction, the output consists of key points marked on the MRI in green color as shown in Fig. 2.

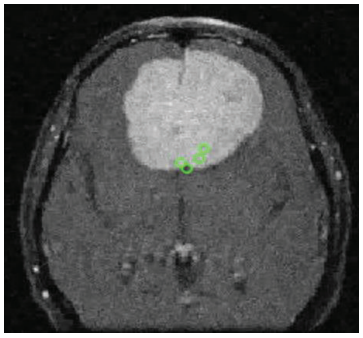

Fig. 2. BRIEF Output- Key points marked on MRI

Multi otsu and Inverse otsu can be good noise to impose on raw images depending on the model we're working with. Out of all the pre - processing techniques implemented, otsu has resulted in the least favourable performance results.

On applying HSV, the validation accuracy improves for the VGG16 model compared to the validation accuracy for VGG16 without pre-processing. However, there is no improvement observed in the other models that were implemented. CNN and VGG16 models have much higher validation accuracies compared to the ML models. Out of all the ML models, the ensemble model which uses KNN as the meta classifier outperforms the accuracies achieved by logistic regression, KNN and SVM. Even though the ensemble model outperformed the ML models, it has underperformed compared to the CNN and VGG16 DL models and performed consistently with all the preprocessing methods.

It is also to be noted that the RESNET50 model gives a significantly lower validation accuracy when any of these pre-processing techniques are applied to the input MRIs.

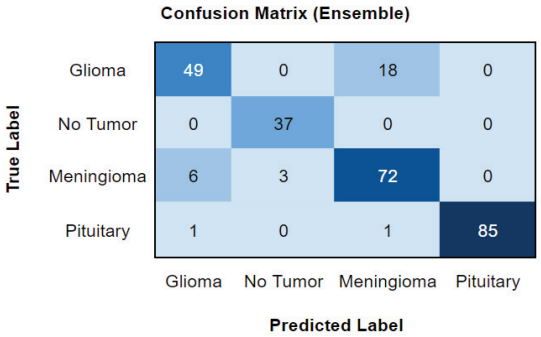

Fig. 3. Ensemble Model Confusion Matrix

Fig. 3, Fig. 4 and Fig. 5 shows the confusion matrix drawn out from the StackingCV Classifier ensemble model to which DWT pre-processed images have been fed, VGG16 model and CNN model to which the original images have been fed respectively.The Confusion matrix in Fig. 6 shows how often the model classified each label correctly (in blue) for the AutoML model without having any preprocessing applied to it.

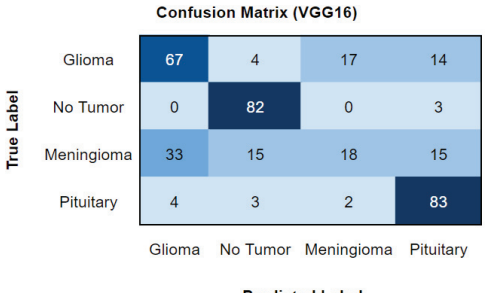

Fig. 4. VGG16 Model Confusion Matrix

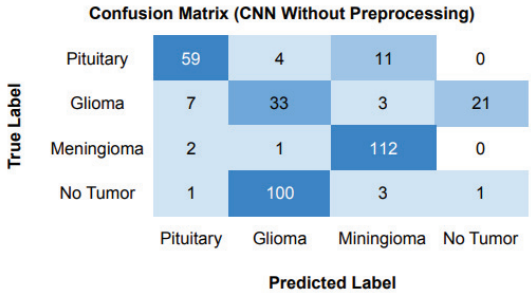

Fig. 5. CNN Model Confusion Matrix

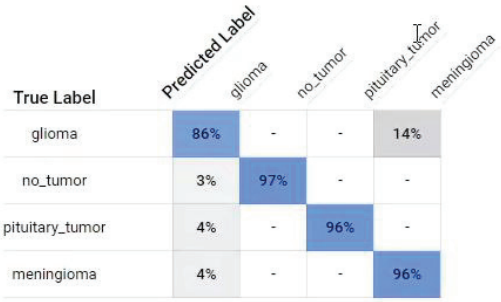

Fig. 6. AutoML Confusion Matrix

Graph in Fig. 7 represents the accuracies of CNN, VGG16, ResNet50, KNN, SVM, Logistic Regression, ensemble model with and without the pre- processing techniques and BRIEF implemented on the MRIs:

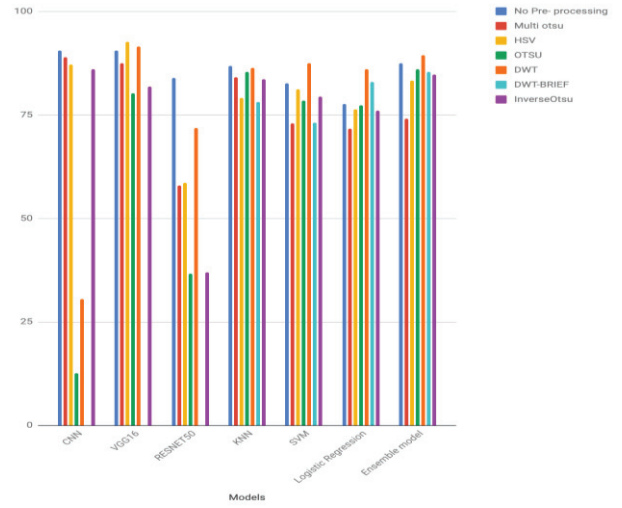

Fig. 7. Graph of all models accuracies

The average precision obtained for the AutoML cloud model is 98.67% and a Recall of 95.68% with an average validation accuracy of 98.702% at a score threshold value of 0.5 when trained for a total of 20 node hours using the Vision API.

While testing the AutoML model the misclassification rate was close to zero for all the four classes with an average precision of 89.23%, 99.39%, 99.69% and 100% without any preprocessing applied to the images whereas the highest accuracy that was obtained for other models was in VGG16 with HSV Preprocessing at 92.68% validation accuracy and average precision of 94.29%, 11.25%, 0.87% and 6.76% for Glioma, Meningioma, Pituitary and No Tumor respectively

V. Conclusion And Future Work

In this paper, various ML and TL models along with different preprocessing techniques are compared with each other. The detected tumor was classified as one of the following four classes viz. glioma, meningioma, pituitary tumor or no tumor. DWT, HSV, Otsu, Inverse Otsu and Multi Otsu are the different pre-processing techniques applied separately on the brain MRI images. These images were given as input to the models with BRIEF descriptor helping us draw out key points present in tumorous MRIs. AutoML has been executed on the brain MRI images to evaluate how efficacious it is in comparison to the other ML and TL models. The work presented in this paper can be extended in detection of more tumor types such as Medulloblastoma, Oligodendroglioma and so on. Moreover, classification of tumors can be extended to other regions of the human body.

References

[1] Bathe, Kavita and Rana, Varun and Singh, Sanjay and Singh, Vijay, Brain Tumor Detection Using Deep Learning Techniques (MAY 7, 2021). Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021), Available at SSRN: https://ssrn.com/abstract=3867216 or http://dx.doi.org/10.2139/ssrn.3867216

[2] Tahia Tazin, Sraboni Sarker, Punit Gupta, Fozayel Ibn Ayaz, Sumaia Islam, Mohammad Monirujjaman Khan, Sami Bourouis, Sahar Ahmed Idris, Hammam Alshazly, "A Robust and Novel Approach for Brain Tumor Classification Using Convolutional Neural Network", Computational Intelligence and Neuroscience, vol. 2021, Article ID 2392395, 11 pages, 2021. https://doi.org/10.1155/2021/2392395

[3] George*, T., & Ramakrishnan, D. T. (2020). A Professional Estimate on the Segmentation of Brain Cancer in MR Images using M-FCM. International Journal of Recent Technology and Engineering (IJRTE), 9(1), 2425–2430. https://doi.org/10.35940/ijrte.a2920.059120

[4] S. Grampurohit, V. Shalavadi, V. R. Dhotargavi, M. Kudari and S. Jolad, "Brain Tumor Detection Using Deep Learning Models," 2020 IEEE India Council International Subsections Conference (INDISCON), 2020, pp. 129-134, doi: 10.1109/INDISCON50162.2020.00037.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[5] A. Hussain and A. Khunteta, "Semantic Segmentation of Brain Tumor from MRI Images and SVM Classification using GLCM Features," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 38-43, doi: 10.1109/ICIRCA48905.2020.9183385.

[6] Anto Bennet, M., Haritha, D., Karthika, P., Mahalakshmi, K., & Pavithra, B. (2019). Identification and Detection of Brain Tumor Segmentation using Fuzzy and Neural Network. International Journal of Recent Technology and Engineering (IJRTE), 7(6S3). https://www.ijrte.org/wp-content/uploads/papers/v7i6s3/F1038376S19.pdf

[7] Somasundaram, S., & Gopinath, R. (2019). A Hybrid Convolutional Neural Network and Deep Belief Network for Brain Tumor Detection in MR Images. International Journal of Recent Technology and Engineering, 8(2S4), 979–985. https://doi.org/10.35940/ijrte.b1193.0782s419.

[8] Hossain, Tonmoy; Shishir, Fairuz Shadmani; Ashraf, Mohsena; Al Nasim, MD Abdullah; Muhammad Shah, Faisal (2019). [IEEE 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) - Dhaka, Bangladesh (2019.5.3-2019.5.5)] 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) - Brain Tumor Detection Using Convolutional Neural Network. , (), 1–6.doi:10.1109/ICASERT.2019.8934561

[9] Kurnar, Mahesh; Sinha, Aman; Bansode, Nutan V. (2018). [IEEE 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Pune, India (2018.8.16-2018.8.18)] 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Detection of Brain Tumor in MRI Images by Applying Segmentation and Area Calculation Method Using SCILAB. , (), 1–5.doi:10.1109/ICCUBEA.2018.8697713

[10] Kumar, T. Sathies; Rashmi, K.; Ramadoss, Sreevidhya; Sandhya, L.K.; Sangeetha, T.J. (2017). [IEEE 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS) - Chennai, India (2017.5.4-2017.5.5)] 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS) - Brain tumor detection using SVM classifier. , (), 318–323. doi:10.1109/SSPS.2017.8071613

[11] Mariam Saii, Zaid Kraitem. Automatic Brain Tumor Detection in MRI Using Image Processing Techniques. Biomedical Statistics and Informatics. Vol. 2, No. 2, 2017, pp. 73-76. doi: 10.11648/j.bsi.20170202.16

[12] Singh, Garima; Ansari, M.A. (2016). [IEEE 2016 1st India International Conference on Information Processing (IICIP) - Delhi, India (2016.8.12-2016.8.14)] 2016 1st India International Conference on Information Processing (IICIP) - Efficient detection of brain tumor from MRIs using K-means segmentation and normalized histogram. , (), 1–6. doi:10.1109/iicip.2016.7975365

[13] Nandi, Anupurba (2015). [IEEE 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS) - Bhubaneshwar, Odisha, India (2015.11.2-2015.11.3)] 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS) - Detection of human brain tumour using MRI image segmentation and morphological operators. , (), 55–60.doi:10.1109/CGVIS.2015.7449892

[14] Telrandhe, Swapnil R.; Pimpalkar, Amit; Kendhe, Ankita (2016). [IEEE 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave) - Coimbatore, India (2016.2.29-2016.3.1)] 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave) -

Detection of brain tumor from MRI images by using segmentation & SVM. , (), 1–6. doi:10.1109/STARTUP.2016.7583949

[15] Sinha, Kailash; Sinha, G. R. (2014). [IEEE 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS) - Bhopal (2014.3.1-2014.3.2)] 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science - Efficient segmentation methods for tumor detection in MRI images. , (), 1–6. doi:10.1109/SCEECS.2014.6804437

[16] Sharma, Yamini; Meghrajani, Yogesh K. (2014). [IEEE 2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN) - Surat, India (2014.12.26-2014.12.27)] 2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking - Brain tumor extraction from MRI image using mathematical morphological reconstruction. , (), 1–4. doi:10.1109/et2ecn.2014.7044982

[17] Abdulbaqi, Hayder Saad; Mohd Zubir Mat, ; Omar, Ahmad Fairuz; Mustafa, Iskandar Shahrim Bin; Abood, Loay Kadom (2014). [IEEE 2014 IEEE Student Conference on Research and Development (SCOReD) - Penang, Malaysia (2014.12.16-2014.12.17)] 2014 IEEE Student Conference on Research and Development - Detecting brain tumor in Magnetic Resonance Images using Hidden Markov Random Fields and Threshold techniques. , (), 1–5. doi:10.1109/SCORED.2014.7072963

[18] Sharma, M. (2012). Brain Tumor Segmentation using hybrid Genetic Algorithm and Artificial Neural Network Fuzzy Inference System (ANFIS). International Journal of Fuzzy Logic Systems, 2(4), 31–42. https://doi.org/10.5121/ijfls.2012.2403

[19] https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri