Topic: Brain Tumor Detection using python, sklearn and OpenCV

Name: Dharan Thaker(40194790) and Arjun Anghan ()

github:-

Abstract:

Image processing has become a critical field in the detection and diagnosis of brain tumors. One of the most challenging tasks in this area is the accurate and efficient classification of digital images of the brain to identify abnormalities. In recent years, machine learning algorithms have emerged as an effective tool for this task, with logistic regression and support vector classifier being two popular methods used in the detection of brain tumors[1]. Logistic regression is a statistical method that is used to analyze relationships between variables. In the context of image processing, logistic regression can be used to classify digital images of the brain into two categories: normal and abnormal. Logistic regression can also be used to extract features from images that can be used to identify abnormalities. Support vector classifier, on the other hand, is a machine learning algorithm that can be used for both classification and regression. In the context of brain tumor detection, support vector classifier can be used to classify digital images of the brain into two or more categories based on their features. In this study, we performed a comparative analysis of logistic regression and support vector classifier for brain tumor detection using image processing. We applied both algorithms to a dataset of digital brain images and evaluated their performance using various evaluation metrics, including accuracy, precision, recall, and F score.

index terms:- Logistic regression, Support Vector Classifier, image classification, accuracy, precision, recall, F score

1. Introduction:

A brain tumor is an abnormal growth or mass of cells in or around the brain. Brain tumors can be either malignant or benign, and they can grow at different rates. While only about one-third of brain tumors are cancerous[2], both cancerous and noncancerous brain tumors can impact brain function and overall health if they grow large enough to press on surrounding nerves, blood vessels, and tissue.

In recent years, artificial intelligence-assisted automatic or semi-automatic detection and segmentation methods have become increasingly important in medical diagnosis for brain tumor detection and segmentation. These methods are critical for medical practitioners to validate the limits and regions of the brain tumor, determine its exact location and affected areas, and plan appropriate treatment such as chemotherapy, radiotherapy, or brain surgery.

In the context of image classification, supervised learning has been used to develop models that can accurately classify images into different categories or classes. Supervised learning is a type of machine learning that involves training a model on labelled data. In supervised learning, the model learns to make predictions by analysing a dataset that contains both input features and their corresponding output labels. The goal of supervised learning is to train a model that can accurately predict the output label for new, unseen input data. Supervised learning has shown great promise in image classification and brain tumor detection, as it allows models to learn from labelled examples to accurately identify patterns in the data that are indicative of specific conditions or features.

These models are trained on a dataset of labelled images that represent each class, allowing the model to learn the features that distinguish between the different classes. In the case of brain tumor detection, supervised learning has been used to develop models that can accurately detect and segment brain tumors from medical images. These models are trained on a dataset of labeled brain images that have been classified as normal or abnormal, allowing the model to learn the patterns that are indicative of a brain tumor.

The first step in making a model to classify brain tumor images is to obtain a dataset that contains images of brain tumors and corresponding labels indicating whether a tumor is present in each image. Once you have obtained a dataset, the next step is to pre-process the images to remove any noise or artifacts that could interfere with the model's ability to accurately classify the images. This might involve techniques such as image filtering or image enhancement to improve the quality of the images.

After pre-processing the images, the next step is to scale the dataset to ensure that all features are on the same scale. This is important because many machine learning algorithms are sensitive to the scale of the features and may perform poorly if features are not normalized. Once the dataset has been scaled, it can be split into a training set and a test set.

The next step is to perform feature selection to identify the most important features that are predictive of whether a brain tumor is present in an image. This involves technique called as principal component analysis to identify the most important features. Once the features have been selected, the next step is to train a machine learning model, such as a logistic regression or support vector classifier, on the training set. Finally, the model can be evaluated on the test set to determine its performance in classifying brain tumor images.

The models that were taken into consideration were Logistic Regression and Support Vector Classifiers (SVC). Once trained, the algorithms can be applied to new, unlabelled images to predict the likelihood that they contain a brain tumor. Logistic regression works by modelling the probability of a brain image being abnormal using a logistic function that maps the image features to the probability of the image being abnormal. On the other hand, support vector classifiers separate the normal and abnormal brain images by constructing a hyperplane that maximally separates the two classes. This hyperplane is determined by the support vectors, which are the images closest to the decision boundary. Both algorithms can be fine-tuned using a range of parameters to improve their accuracy and reduce the risk of overfitting.

In Summary ,this paper identifies the most suited model to the problem of brain tumor classification of images. The subsequent sections in the paper aims to provide the in depth information about other related works, the dataset used, the proposed methodology, the results acquired and an interpretation of these in the conclusion.

1. Related Work

When developing a machine learning-based intelligent system for brain tumor classification, researchers typically begin by segmenting brain tumors through the use of different methods, followed by classification[3]. Although this approach enhances accuracy, it can be time-consuming and requires an additional step prior to network training. Despite this, several researchers have employed CNNs to directly classify brain tumors, bypassing the need for segmentation.

Justin et al. [4] utilized three classifiers - random forest (RF), a fully connected neural network (FCNN), and a CNN - in order to enhance classification accuracy. The CNN demonstrated the highest accuracy rate at 90.26%. Meanwhile, Tahir et al. [3] examined several preprocessing techniques to improve classification results. They experimented with three techniques: noise reduction, contrast enhancement, and edge detection, testing various combinations on different test sets. Their findings suggest that utilizing multiple techniques is more advantageous than relying on one. Tahir et al. employed the Figshare dataset and tested the SVM classifier on it, achieving an 86% accuracy rate. Ismael et al. [5] proposed a new approach to brain tumor classification by combining statistical features with neural networks. They utilized MR images to extract statistical features for classification and incorporated 2D discrete wavelet transforms (DWT) and Gabor filters for feature selection. The segmented MR images were fed into their algorithm, which resulted in an average accuracy of 91.9%. By utilizing the DWT and Gabor filters, they were able to effectively reduce the dimensionality of the features, while still preserving the relevant information. This method proved to be successful in improving the classification accuracy of brain tumors.

In [6], a project aimed to classify multi-grade brain tumors is described. The method uses a pre-trained CNN model in combination with segmented images. To handle class imbalance and improve accuracy, data augmentation is performed using various techniques. The model is validated on three different datasets, and the results presented are compelling.

In [7], Nayoman et al. investigated the use of CNNs for brain tumor classification. They constructed seven different neural networks and found that a lightweight model performed best. Without any prior segmentation, this simple model achieved a test accuracy of 84.19%.

Guo et al. [8] proposed a classifier for Alzheimer's disease using graph convolutional neural networks (GCNNs). The ADNI dataset was used to classify Alzheimer's disease into 2 and 3 categories. The proposed graph nets achieved an accuracy of 93% for 2-class classification compared to 95% for ResNet architecture and 69% for the SVM classifier. In the three-class classification, the proposed graph CNN achieved 77%, ResNet achieved 65%, and SVM achieved 57%.

GAN-based networks have become increasingly popular for generating synthetic medical images due to their remarkable performance. Liu et al. [9] have proposed a variation of Cycle GAN that uses a domain control module (DCM) and a Pseudo Cycle Consistent module (PCCM) to generate Computed Tomography (CT) images. This approach has shown promising results in producing high-quality synthetic CT images that are useful for a variety of medical applications. The DCM allows the user to control the appearance of the generated images, while the PCCM helps to ensure consistency between the generated images and the real ones.

The utilization of GANs has garnered significant interest from researchers, and these models are now being extensively used in various medical imaging domains. Researchers are trying to enhance their results by utilizing more complex and deeper architectures. Although GAN-based studies have different approaches, they all use random Gaussian noise as the input to the generator of the GAN. However, in the field of generative medical imaging, exploring the manipulation of the input noise of GANs is still largely unexplored.

3) Proposed Methodology

[A] Dataset

The dataset is taken from Kaggle is of contains 3264 images from 4 different classes of tumor class namely glioma\_tumor, meningioma\_tumor, no\_tumor, pituitary\_tumor, which has 100 files, 115 files, 105 files and 74 files respectively. Since the images did not have uniform dimensions, so they must be pre processed before hand in order to be applied to the model.

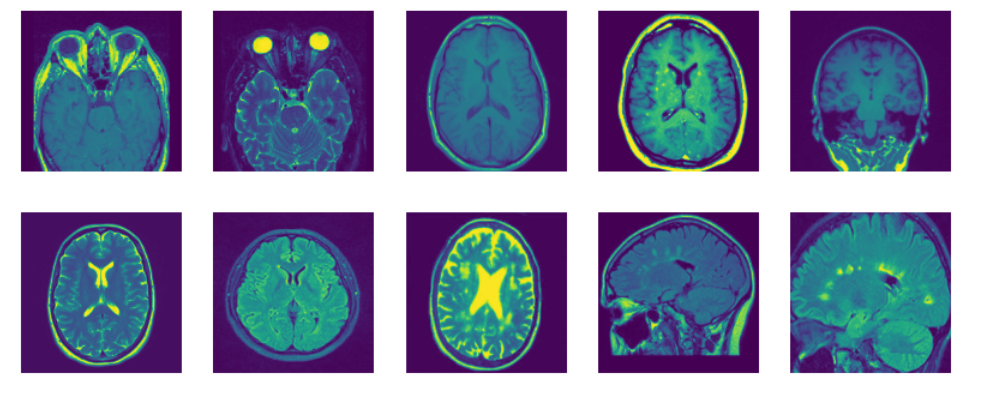


Fig 1. sample images of no tumor and pituitary tumor

[B] Preprocessing

The data that we have is still raw and inconsistent and it can have misleading results if it is directly applied to the model. So pre-processing is required to make the set clean and normalized to improve the quality of the dataset.

Image Reshaping: The numpy array needs to be reshaped by changing its dimensions without changing its data, this allows to manipulate the dimensions of these arrays to perform operations. Specifically, it reshapes into a 2D numpy array where the number of rows is equal to the length of array and the columns is determined automatically based on the size of the original data in the array. The ‘-1’ in the reshape function is a placeholder for the number of columns that needs to be determined automatically based on the size of the original data in the array. By using ‘-1’, numpy automatically calculates the correct number of columns such that the total number of elements in the reshaped array remains the same as in the original array.

[C] Splitting the dataset

Splitting a dataset is a common technique in machine learning for evaluating the performance of models and preventing overfitting. The basic idea is to divide the dataset into separate training and testing sets, where the training set is used to train the model, and the testing set is used to evaluate its performance. Here the splitting is done with the random state of 10 and the ratio of the training and testing dataset is kept to be 80:20. After the splitting of the dataset, the shape of the x\_train and x\_test came out to be ((977,40000)) and ((245,40000)) respectively.

[D] Data Balancing

## put some of images of data balancing

[E] Feature Scaling

feature scaling refers to the process of scaling or standardizing the input features of a dataset so that they are on the same scale. This is done to ensure that no single feature dominates the learning algorithm and to prevent the influence of features with larger scales on the model's outcome. The normalization of pixel values between 0 and 1 is an example of feature scaling. Here, we have used the min-max feature scaling technique to scale the pixel values of the image between 0 and 1. First, the maximum and minimum pixel values of the testing data are printed using the ’max()’ and ‘min()’ functions of Numpy. Next, the pixel values of both the training and testing data is divided by 255, which is the maximum pixel value that can be represented by an 8-bit image. This results in pixel values that are now between 0 and 1. Finally, the maximum and minimum pixel values of the normalized training and testing data are printed again using the ‘max()’ and ‘min()’ functions of numpy. This step useful to verify that the min-max scaling has been applied correctly, and the pixel values are now between 0 and 1.

[F] Feature Selection: Principal Component Analysis (PCA)

PCA (Principal Component Analysis) is a technique for feature selection and dimensionality reduction in machine learning. Feature selection involves selecting a subset of the most relevant features from the original dataset, while dimensionality reduction involves transforming the original features into a smaller set of new features, without losing much of the information content. In PCA, the goal is to transform the original dataset into a new set of uncorrelated variables called principal components. Here, we have created a PCA object with the desired level of explained variance of 0.98. This means that PCA will keep enough principal components to explain 98% of the variance in the training data, while discarding the remaining components that contribute less to the overall variance.

[G] Dependencies

The project uses dependencies like pandas which is used for the cleaning and manipulation of dataset, Numpy which is used for working with images as they are 2D arrays, and numpy is a python library which is used for working with arrays. In addition to that, OpenCV is used for visualizing the images on different scales and sizes, as it is the library meant for handling and manipulating the images. Sklearn is used for performing various tasks like feature selection using PCA, training the model over logistic regression and Support Vector Classifiers by importing the in-built functionalities provided by Sklearn.

[H] Model Training

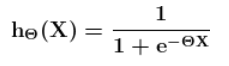
There are two model architecture that have been considered in this task of Brain tumor Classification. The two models are Logistic Regression and Support Vector Classifier, and a comparative analysis is done between these two models and the model that was better in terms of the evaluation metrics was considered for further classification of images.

Logistic regression and support vector classifiers are both supervised learning algorithms that have shown great promise in the detection and segmentation of brain tumors. These algorithms work by identifying patterns or features in labeled datasets that are used to classify new, unlabeled data. In the case of brain tumor detection and segmentation, the algorithms are trained on a dataset of brain images that have been labeled as either normal or abnormal. The algorithms analyze the labeled data to identify patterns that distinguish between normal and abnormal brain images. The identified patterns can include variations in texture, shape, and intensity of image pixels.

Logistic Regression is used when the dependent variable(target) is categorical [10]. For example,

* To predict whether an email is spam (1) or (0)
* Whether the tumor is malignant (1) or not (0)

The output of logistic regression is always between (0, and 1), which is suitable for a binary classification task. The higher the value, the higher the probability that the current sample is classified as class=1, and vice versa.



As the formula above shows, Equation is the parameter we want to learn or train or optimize and Equation is the input data. The output is the prediction value when the value is closer to 1, which means the instance is more likely to be a positive sample(y=1). If the value is closer to 0, this means the instance is more likely to be a negative sample(y=0)[11].

In practice, the [logistic regression algorithm](https://www.interviewbit.com/data-science-interview-questions/) analyzes relationships between variables. It assigns probabilities to discrete outcomes using the Sigmoid function, which converts numerical results into an expression of probability between 0 and 1.0. Probability is either 0 or 1, depending on whether the event happens or not. For binary predictions, you can divide the population into two groups with a cut-off of 0.5. Everything above 0.5 is considered to belong to group A, and everything below is considered to belong to group B[11].

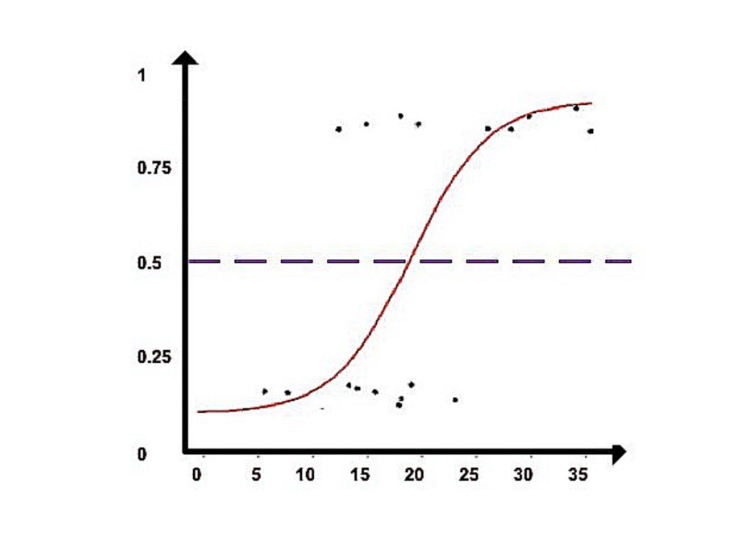


Fig. 2 hyperplane separating two categories

According to the SVM algorithm we find the points closest to the line from both the classes. These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane[12].

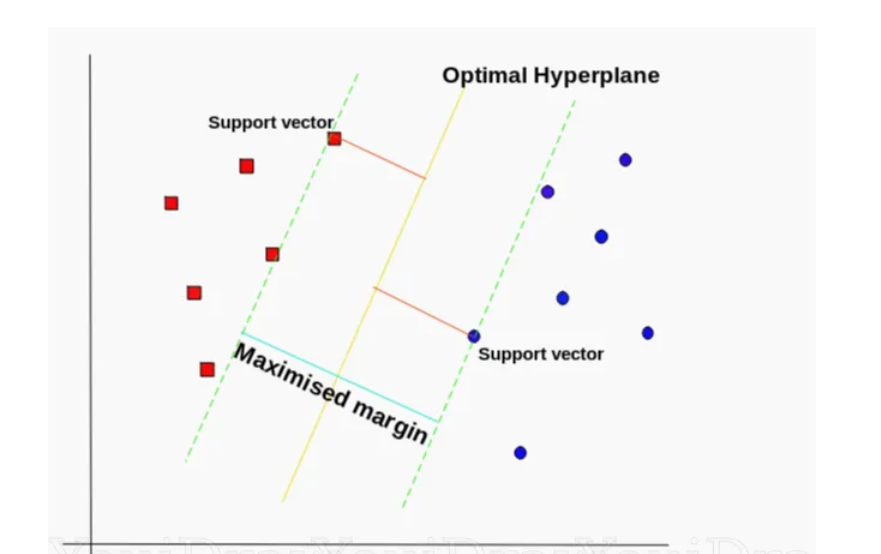


fig. 3 Optimal Hyperplane using the SVM algorithm

Thus SVM tries to make a decision boundary in such a way that the separation between the two classes (that street) is as wide as possible. SVM works well with unstructured and semi-structured data like text and images while logistic regression works with already identified independent variables.

Sklearn library has been used in order to import logistic regression and support vector classifier models. After training both the models, it was deduced that SVC model performed better in comparison to the logistic regression model. The reason is because SVM is based on the geometric properties of the data while logistic regression is based on the statistical approaches of the data. Thus, majority of the image classification problems are better solved with SVC than with logistic regression.

1. Evaluation and Results

[A] System environment

Both the models were trained on a medium tier system in terms of the computational power. The specifications of the system are as follows:

CPU: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz

GPU: Intel(R) UHD Graphics 620

Installed RAM: 8.00 GB (7.89 GB usable)

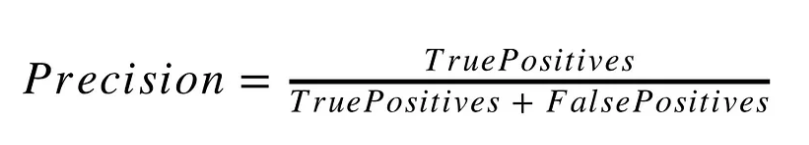
[B] Evaluation Criteria

Evaluation metrics are used to measure the performance of a machine learning model and determine how well it can predict outcomes. Here are some common evaluation metrics used in machine learning and model training.

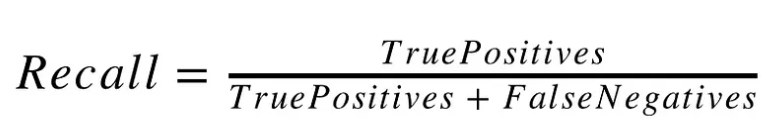
1. Accuracy: Accuracy measures the proportion of correct predictions made by the model. It is calculated by dividing the number of correct predictions by the total number of predictions.



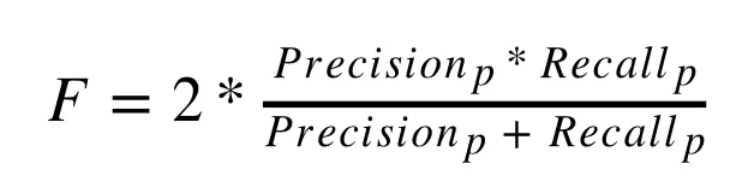
1. Precision: Precision measures the proportion of true positives (correctly predicted positive instances) out of all instances that were predicted as positive. It is calculated by dividing the number of true positives by the sum of true positives and false positives.



1. Recall: Recall measures the proportion of true positives out of all actual positive instances. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.



1. F score: The f score is a harmonic mean of precision and recall. It balances the trade-off between precision and recall and is often used when both are important.



After training both the models, we found that the evaluation metrics of SVM came out to be better than the logistic regression.

[C] Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a classification model by comparing its predicted output to the actual output. The matrix shows the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class in the classification.

The confusion matrix for logistic regression model is shown in the figure [4] below:

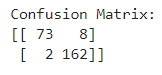


Fig. 4 Confusion matrix for logistic regression

The confusion matrix for SVC model is shown in the figure [5] below:

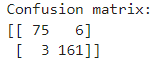


Fig. 5 Confusion matrix for SVC

After training the model in SVC, the predictions obtained for the test cases of non tumor datasets is shown in the figure below:

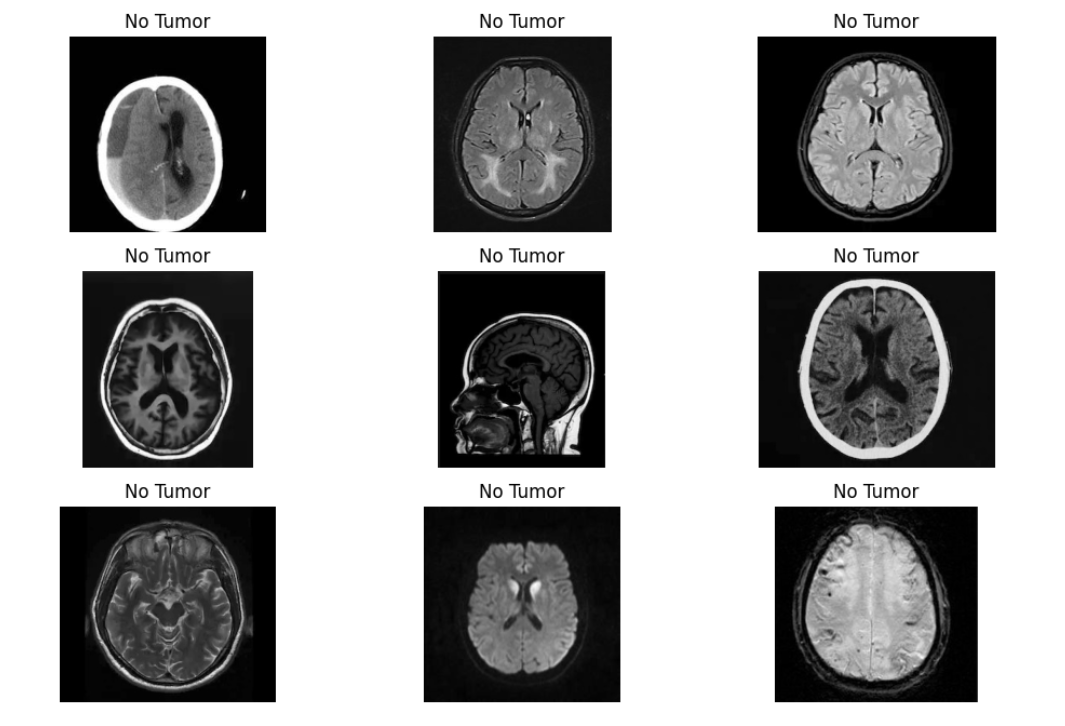


Fig 6. Predictions on test data for no tumor

The predictions obtained on the test data with regards to pituitary\_tumor is shown in the figure below:

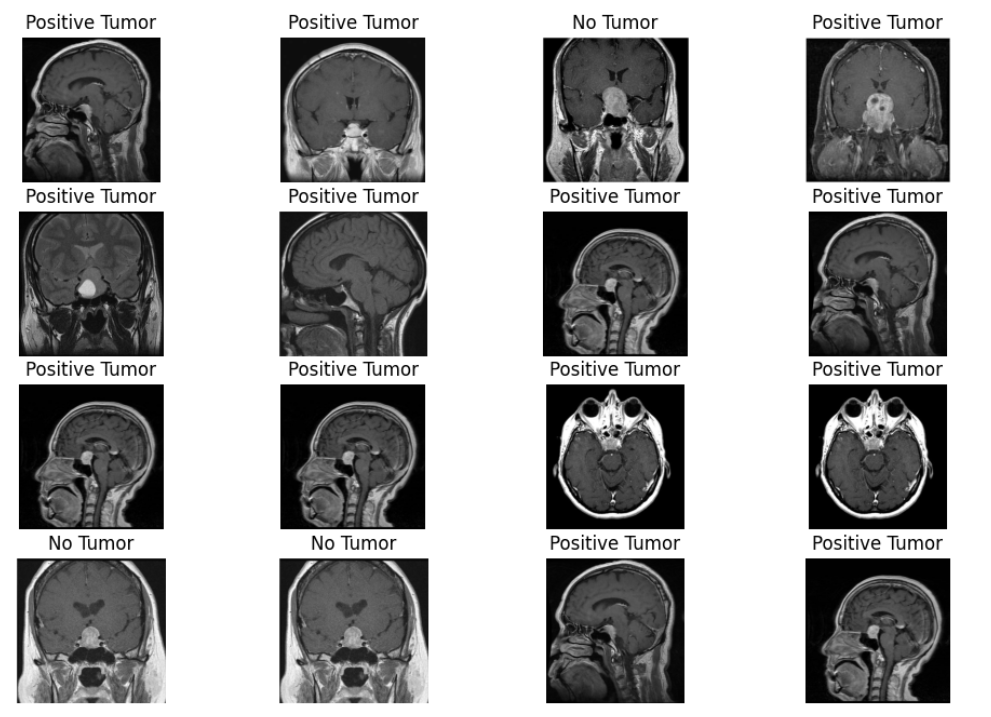


Fig 7. Predictions on test data for pituitary\_tumor

Figure[6] shows that for all the test cases in non tumor, the SVC model gave the right results and Figure[7] showed that test cases involving the pituitary\_tumor, 13 out of 16 images were predicted right.

[D] Results

The results obtained after applying two different models that are logistic regression and support vector classifier concluded that the SVC model performed relatively better than the logistic regression model. This holds true for the evaluation metrices imported from sklearn library. The metrices that are considered are accuracy, precision, recall and f-score. The results can be seen in the figures [8] and [9] below:

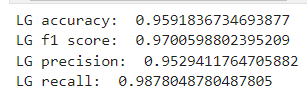


Fig 8. Evaluation Metrices for Logistic Regression

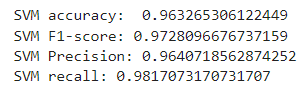


Fig 9. Evaluation metrices for Support Vector Classifier

SVC came out to be better than logistic regression because it was able to handle the non-linear classification boundaries more effectively. Logistic regression models can only fit linear decision boundaries, which may not be suitable for more complex and non -linear datasets like images. This was possible because of the technique that SVC uses called the kernel trick, which maps the input data into a higher dimensional feature space where linear separation can be performed more effectively. This in turn allowed the SVC to model complex decision boundaries that can better separate different classes in the image.

Top of Form

1. Conclusion and Future Work

The Support Vector Classifier model turned out to be a better approach for the brain tumor classification problem which was the problem statement considered in this paper. The SVC model performed relatively well in terms of the evaluation metrices resulting in high accuracy and high f score than the logistic regression model.

Although in terms of the future work that could be done in brain tumor classification is the use of Convolutional Neural Networks (CNNs) which are a popular type of deep learning model that have shown great success in medical image analysis, including brain tumor classification. CNNs are designed to automatically learn spatial features from images, making them particularly well-suited to medical image analysis tasks. As several studies have reported high accuracy using deep learning approaches for brain tumor classification. For example, a study published in the journal[13] of digital imaging in 2019 reported much higher accuracy than other models for classification of brain tumors using a CNN-based model.

References

1) <https://www.sciencedirect.com/topics/computer-science/tumor-detection>

2) <https://my.clevelandclinic.org/health/diseases/6149-brain-cancer-brain-tumor>

3) Tahir B., Iqbal S., Usman Ghani Khan M., Saba T., Mehmood Z., Anjum A., Mahmood T. Feature enhancement framework for brain tumor segmentation and classification. *Microsc. Res. Tech.*2019;82:803–811. doi: 10.1002/jemt.23224.

4) Paul J.S., Plassard A.J., Landman B.A., Fabbri D. Deep Learning for Brain Tumor Classification. *Proc. SPIE Spie Med. Imaging .*2017;10137:1–16.

5) Ismael M.R., Abdel-Qader I. Brain Tumor Classification via Statistical Features and Back-Propagation Neural Network; Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT); Rochester, MI, USA. 3–5 May 2018; Piscataway, NJ, USA: IEEE; 2018. pp. 252–257.

6) Sajjad M., Khan S., Muhammad K., Wu W., Ullah A., Baik S.W. Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *J. Comput. Sci.*2019;30:174–182. doi: 10.1016/j.jocs.2018.12.003.

7) Abiwinanda N., Hanif M., Hesaputra S.T., Handayani A., Mengko T.R. *World Congress on Medical Physics and Biomedical Engineering 2018.* Springer; Singapore: 2019. Brain Tumor Classification Using Convolutional Neural Network; pp. 183–189.

8) Guo J., Qiu W., Li X., Zhao X., Guo N., Li Q. Predicting Alzheimer’s Disease by Hierarchical Graph Convolution from Positron Emission Tomography Imaging; Proceedings of the 2019 IEEE International Conference on Big Data (Big Data); Los Angeles, CA, USA. 9–12 December 2019; Piscataway, NJ, USA: IEEE; 2019. pp. 5359–5363.

9) Liu Y., Chen A., Shi H., Huang S., Zheng W., Liu Z., Zhang Q., Yang X. CT synthesis from MRI using multi-cycle GAN for head-and-neck radiation therapy. *Comput. Med. Imaging Graph.*2021;91:101953. doi: 10.1016/j.compmedimag.2021.101953.

10) <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>

11) <https://www.kdnuggets.com/2022/07/logistic-regression-work.html#:~:text=is%20Logistic%20Regression%3F-,Logistic%20regression%20is%20a%20Machine%20Learning%20classification%20algorithm%20that%20is,the%20logistic%20of%20the%20result>.

12) <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>

13) https://www.springer.com/journal/10278