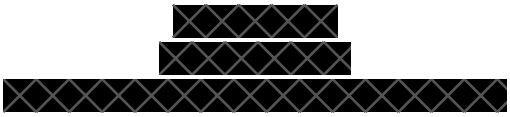
IT1244 Project Report: Brain Tumor Detection

Team 6
Malika Eden D'Sa



Introduction

Brain tumours are abnormal growths of tissues within the brain that can interfere with normal brain function. There are two types of brain tumours, malignant, i.e., cancerous, and benign, i.e., non-cancerous. It often takes a long time and relies on a radiologist's abilities and experience to diagnose a brain tumour. With a large number of patients, this process can be all the more tedious. In such situations, machine learning can help save an enormous amount of resources in a life and death situation. This report will give an overview of the models used to classify an MRI image of a brain tumour into "malignant" or "benign" classes.

Some recent models used in the classification of brain tumours as malignant and benign include using Support Vector Machine (SVM) classifier (Malarvizhi et al.) and a classification model based on Convolutional Neural Network (CNN) model (Hanaa ZainEldin et al.).

Some of the drawbacks in these research papers were the high processing time due to optimization of hyperparameters in the CNN model and limited training data. For the SVM classifier, the overall accuracy rate was up to 80%. Given the critical nature of brain tumour classification tasks, it is essential to achieve higher accuracy levels.

Dataset

The data folder itself consists of two folders "benign" and "malignant" and both contain Magnetic Resonance Imaging (MRI) images that are saved as JPG files of the two types of brain tumours respectively. The "benign" folder consists of 77 MRI images while the "malignant" folder consists of 154 MRI images.

There were a few issues with the dataset itself. Firstly, both folders do not contain the same number of MRI images. This imbalanced data might lead to the model becoming more biassed towards the majority class – which is malignant. In addition, this might lead to poor generalisation of the minority class due to fewer images to learn from which may result in lower accuracy, precision, recall or F1-score for the minority class – which is benign. To tackle this issue, image augmentation was attempted for the benign class in order to obtain the same number of images in both classes. This is further explained in our methodology section.

Another issue in the dataset is that in all of the MRI images, there is a huge black frame presented regardless of the class. This may lead to feature distortion where the black frame introduces additional noise or irrelevant features into the images, making it harder for the model to extract meaningful information. This could lead to reduced model performance. To tackle this issue, we attempted pre-processing the images where the black frames were cropped out from all the images before feeding them into the model.

Lastly, the dataset may contain both RGB and grayscale images. If both types of images are fed through the model without processing, this might lead to input inconsistency as RGB images have three colour channels (red, green, blue), while grayscale images have only one channel representing intensity. To tackle this issue, we ensured that only RGB images were considered during our data preparation. However, it was found that this particular dataset does not have any grayscale images.

Methodology

In this report, we consider both classical machine learning and deep learning methods. The reason for this was to have a more comprehensive and informed discussion on which model was the best. The classical machine learning methods used for the classification task are K-Nearest Neighbours (KNN), Logistic Regression, Random Forest Classifer and Support Vector Machine (SVM). The last two are not covered in IT1244. The deep learning methods used are Convolution Neural Network (CNN) and Artificial Neural Network (ANN). These 6 models were chosen due to their frequent application in classification tasks and the details of them are covered under the subsection "Model".

The dataset was first divided into training data and testing data in an 80:20 ratio with the 'train test split' method. Stratified split was used to ensure both the train and test data have the same proportion of malignant and benign images. Then we did some exploratory data analysis.

Exploratory Data Analysis

After loading the data, the images first need to be resized so that we can concatenate them. However, resizing images

may result in loss of information, hence, to choose our image size, we plotted the following graph to see the different image sizes of our dataset.

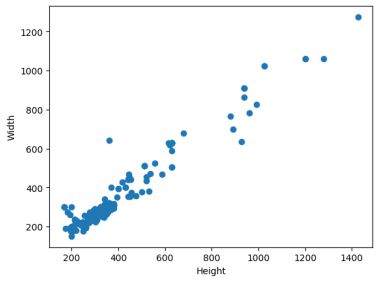


Figure 1: Dimension of images in dataset

From Figure 1, it can be seen that the image sizes are clustered in the range 200-400 px. We then chose (224, 224) as our image size because (1) this size falls in the range 200-400 px where most data points are clustered at, (2) famous pre-training schemes such as VGG19 use this image size as input size as can be seen in the paper by Mostafiz et al.

Feature Extraction

To extract features to be used for fitting our models from images may not be so straightforward. Firstly, we need to understand that images are stored as a matrix of pixel values ranging from 0 to 255 which represent the intensity. For coloured images, the standard RGB contains 3 "panels" of such matrices and it is represented as a tensor. We have explored various feature extraction methods in this project:

1. Flattening the image data into 1 dimensional vector

   This is a naive method of extracting features for image data. While this method is simple, it might not capture spatial relations of the image, meaning that if the malignant tumour is in another location of the brain, the feature may be completely different.

2. Colour histogram

   According to Swain and Ballard, colour histograms count the number of times a given intensity (pixel values) occur in an image. This image feature helps capture spatial relation of the image but may not work so well for object (tumour) detection as shapes and edges are more important rather than colour of the image. We utilised openCV built-in function (calcHist) to compute the colour histogram.

3. Histogram of Oriented Gradients (HOG)

   HOG is a famous image feature used in many image detection tasks due to its ability to detect shapes and edges accurately using local intensity gradients. Briefly, gradients are calculated by considering the changes in pixel values / intensity in a localised region in the x and y directions, and orientations can be obtained easily given the gradients in x and y directions. HOG then computes the histogram of gradient orientations and this is subsequently used for object detection based on the gradient orientations distributions. We utilised skimage libraries to calculate HOG (Dalal and Triggs). Figure 2 obtained from Mamedov et al. demonstrates the oriented gradients of a brain MRI image.

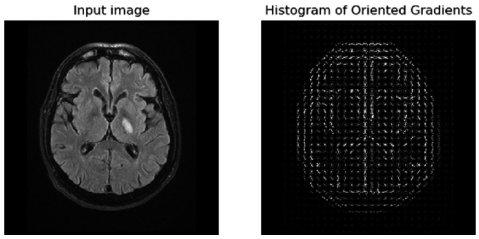


Figure 2: Histogram of oriented gradients of brain MRI

4. Features extracted from pre-trained VGG19 model (transfer learning)

   As covered in the lecture, CNN is capable of learning various features of an image by itself. As such, huge CNN architecture such as VGG19 with pre-trained weights (based on imagenet data) can help us extract high level features of our brain tumour images. This feature extraction method of using a pre-trained model is called transfer learning and was mentioned in Tutorial 9. In this project, we have chosen VGG19 architecture for feature extraction as it was shown to outperform other CNN architectures in the paper by Dalal and Triggs. The fusion of deep learning methods to extract and generate features and classical machine learning methods to train a model is widely used especially when we have a small dataset (Dalal and Triggs & Mostafiz et al.).

Data Augmentation

We augmented the training data by horizontally flipping random images labelled "benign" to match the number of "malignant" images to see if it improved the performance of the model. Only horizontal flipping was used to double the number of benign images to match the number of malignant images. As the MRI images in the dataset come with a "standard" format, performing rotation and vertical flipping of the image may not make much sense. After augmentation, our training set contains 122 benign data and 123 malignant data which is now balanced.

Cropping the Image

As mentioned in Dataset section, we performed some image processing adapted from Rosebrock. More details can be found in our code file.

Evaluation Metrics

In the context of tumour detection, it is crucial to minimise false negatives as failure to alert a case of malignant tumour can impose severe health consequences for a patient (here we consider malignant tumour to be positive). Hence, we are largely concerned with ensuring our model is able to correctly identify malignant tumours with minimal false negatives.

The recall score has the formula $Recall = \frac{TP}{TP+FN}$ and captures the effectiveness of the model to correctly identify all true positive instances, in this case, malignant tumours. Hence, we can seek to maximize recall of the model (we do not want to misclassify true positive instances as negative). However, a model can always predict all points to be positive and achieve a recall score of 1.00.

Therefore, we consider the F1-score that also takes precision into account. Precision has the formula $Precision = \frac{TP}{TP+FP}$ and shows how correct our models are in predicting a positive class. Since F1 score has the formula $F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$, it allows us to have a more balanced viewpoint of how our models perform in terms of both precision and recall. We did not consider accuracy as it has been mentioned that it is not a good evaluation metric.

Cross Validation

When implementing our models, 5-fold cross validation (CV) was done on the training set and average F1 score was used as the main performance metric. The model was then tested on the original test dataset from the first split (that was not touched at all previously) and the classification report was obtained. The average F1 score acquired from 5-fold CV allows us to evaluate the robustness of the model across different datasets, while the final performance metrics allow us to assess the performance of the model on unseen data. Figure 3 adapted from scikit-learn shows how 5-fold CV works in terms of dividing the dataset for each iteration. Note that the error is also reduced as it has been spread across the 5 iterations.

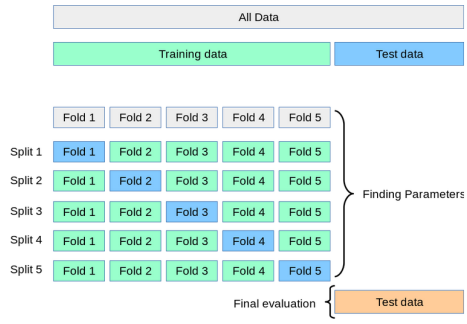


Figure 3: K-fold validation (Scikit-Learn)

Model

As mentioned, we will use 4 classical ML models in this project and they are KNN, Logistic regression, Random Forest Classifier and Support Vector Machines. As the first two have already been covered in lecture, we omit the details of these algorithms in the report. We referred to Bishop's textbook "Pattern Recognition and Machine Learning" to explain the last two algorithms.

1. Decision Trees & Random Forest Classifier

   Decision tree is a supervised learning algorithm that recursively splits the feature space into smaller segments based on the values of the input features, whereby each split is determined with the goal of maximising information gain or minimising entropies. A final prediction is then made at the leaf nodes by majority vote.

   Random forest classifier is an ensemble learning method that is built on multiple decision trees trained on a random subset of data using a random subset of input features. A final prediction is then made by a majority vote based on the predictions made by all trained decision trees. This classifier utilises the technique of bagging that aims to minimise variance of decision trees (Bishop).

2. Support Vector Machine (SVM)

   SVM is a famous supervised learning algorithm that separates the data points using a hyperplane that maximises the margin between the classes. It can be used to separate both linear and non-linear data by mapping the data to higher-dimensional feature space using a kernel function. The SVM formulation is quite mathematically involved and is thus omitted in this report (Bishop).

On top of the 4 classical ML models, we also tried to use deep learning models such as Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) that were taught in lectures. We will utilize the scikit-learn libraries for the implementation of classicla ML algorithms and Tensorflow Keras for the implementation of deep learning models.

To summarise our methodology, after splitting the dataset 80/20 into training and testing sets, exploratory data analysis (feature extraction, augmentation, and cropping) was done on the training set and the non-deep learning models were trained through 5-fold CV. Then the models were evaluated with the final unseen test set data. Then we performed a 5-fold CV with our deep learning models and evaluated them with the final unseen test set data.

Results & Discussion

The performance of the classical ML models has been summarized in Table 1. CV F1 in the table refers to the average F1 score obtained from cross validation and Test F1 in the table refers to the F1 score obtained from the test dataset. A few observations and inferences can be made from the results.

1. We can see that the model performance (in terms of F1 score) is higher when the models are trained with HOG and VGG features. This to be expected and consistent with our understandings of the features. According to the paper by Dalal and Triggs, HOGs are excellent at identifying the edges and shapes of the image and thus performed well in identifying malignant tumours based on the shapes and edges of the tumours in the images.

   On the other hand, VGG features have been pre-trained with the imagenet dataset containing 14 million images using a deep CNN architecture (VGG19). Hence, the high-level features learned from this pre-trained model would describe the image a lot better than the other features considered in this project. However, one drawback

Pre-Processing	Classifier	Feature							
		Flattened		Color Histogram		HOG		VGG	
		CV F1	Test F1	CV F1	Test F1	CV F1	Test F1	CV F1	Test F1
Without Pre-Processing	KNN	0.74	0.79	0.79	0.84	0.82	0.89	0.56	0.73
	Logistic Regression	0.80	0.79	0.82	0.85	0.88	0.89	0.91	0.89
	Random Forest	0.87	0.81	0.86	0.78	0.89	0.91	0.90	0.92
	SVM	0.84	0.88	0.83	0.83	0.88	0.87	0.89	0.85
With Cropping	KNN	0.67	0.54	0.82	0.80	0.63	0.69	0.77	0.76
	Logistic Regression	0.80	0.79	0.75	0.81	0.85	0.91	0.89	0.97
	Random Forest	0.83	0.78	0.82	0.74	0.83	0.81	0.87	0.97
	SVM	0.80	0.79	0.83	0.79	0.84	0.90	0.87	0.83
With Augmentation	KNN	0.61	0.60	0.68	0.83	0.76	0.68	0.14	0.32
	Logistic Regression	0.72	0.77	0.75	0.87	0.83	0.88	0.86	0.90
	Random Forest	0.80	0.81	0.78	0.78	0.83	0.90	0.87	0.94
	SVM	0.72	0.78	0.69	0.84	0.82	0.90	0.86	0.90

Table 1: Classical ML Models Performance on our Dataset

of this feature is that we do not really understand how the network learned the features and thus it is not explainable.

2. Surprisingly, dataset with pre-processing (cropping and augmentation) done did not perform better than the dataset without any pre-processing. In fact, the difference of their CV F1 score and test F1 score is also higher, indicating poorer generalization of the models.

   One possible explanation to this result could be that when the images are cropped, some information may be lost, for instance, small part of the edges may be cut off. Since our features could be robust to the noise of the black space, cropping the image only reduces information without improving the performance.

   Whereas for data augmentation, it could be that the augmented image looks rather similar to the original image, resulting in having two very similar data points in the training set. This could have led to overfitting of the models where the models try to memorize the instances of benign data.
3. KNN models did not perform as well as compared to other models. This can be explained by how KNN is a distance-based algorithm and suffers from curse of dimensionality. Briefly, more and more data points have the same pair-wise distances when the dimension of the features increases, and since our feature dimensions are quite large, we will expect KNN to not perform well.
4. Comparing the performances of different models, it seems that Random Forest Classifier performs the best. This is no surprise to us as ensemble learning method tends to generalize well while attaining higher accuracy.

Next, the performance of ANN and CNN is summarized below. Note that since cropping and data augmentation did not improve the performance of our classical machine learning models, and in the interest of optimising our workflow and maximising efficiency, we did not apply these techniques for our deep learning models.

ANN with Flattened data: CV F1 = 0.80, Test F1 = 0.82
ANN with Color Histogram: CV F1 = 0.81, Test F1 = 0.89
ANN with HOG: CV F1 = 0.89, Test F1 = 0.90
ANN with VGG Features: CV F1 = 0.91, Test F1 = 0.89
CNN: CV F1 = 0.79, Test F1 = 0.86

We can see that the performance of neural networks with our features worked quite well, achieving similar F1 scores as our best classical ML model (Random Forest). Moreover, we did not tune the hyperparameters of our neural network (number of layers, number of nodes, dropout rate, regularization) that much as it would take us a long time to achieve a much better result. Hence, we believe that if we tune the hyperparameters, neural network model performance will be better than classical ML models. In fact, this could be a consideration for future work.

Besides, as our understanding of neural network is not that deep, we do not choose this as our final model as it is not explainable - we do not know how the network learn the features. This thus leads to our conclusion below.

Conclusion

From the discussion above, we concluded that random forest classifier is our best model with a remarkable test F1 score of 0.89 and CV F1 score of 0.91 using HOG features without image pre-processing and data augmentation. While the performance of random forest classifier using VGG features is higher, we decided not to use that in our final model as preparing the feature may take a much longer time as compared to HOG. Moreover, it also utilizes deep learning network which is not very explainable.

While we believe that deep learning network can perform better with more hyperparameter tuning, we also decided to not use it as explained in our discussion above.

References

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886–893.

[Malarvizhi et al., 2022] Malarvizhi, A. B., Mofika, A., Monapreetha, M., and Arunnagiri, A. M. (2022). Brain tumour classification using machine learning algorithm. *Journal of Physics: Conference Series*, 2318:012042.

[Mamedov et al., 2021] Mamedov, N., Kulikova, S., Drobakha, V., Bartuli, E., and Ragachev, P. (2021). *Automatic Segmentation of Acute Stroke Lesions Using Convolutional Neural Networks and Histograms of Oriented Gradients*, pages 205–211.

[Mostafiz et al., 2021] Mostafiz, R., Uddin, M. S., Alam, N.-A., Hasan, M. M., and Rahman, M. M. (2021). Mri-based brain tumor detection using the fusion of histogram oriented gradients and neural features. *Evolutionary Intelligence*, 14:1075–1087.

[Rosebrock, 2016] Rosebrock, A. (2016). Finding extreme points in contours with opencv.

[SciKit-Learn, 2009] SciKit-Learn (2009). 3.1. cross-validation: evaluating estimator performance — scikit-learn 0.21.3 documentation.

[Swain and Ballard, 2002] Swain, M. J. and Ballard, D. H. (2002). Color indexing received january 22, 1991. revised jane 6, 1991. *Elsevier eBooks*, pages 265–277.

[ZainEldin et al., 2022] ZainEldin, H., Gamel, S. A., Elkenawy, E.-S. M., Alharbi, A., Khafga, D., Ibrahim, A., and Talaat, F. M. (2022). Brain tumor detection and classification using deep learning and sine-cosine fitness grey wolf optimization. 10:18–18.