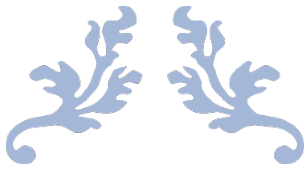


COMP9417 ML VS. CANCER

Group Name: Five Brother

Kairui Jin z5300555

Yanjun Mu z5273254

Junjing Yu z5338098

Wenjie Jiang z 5301816

Mengzhen Li z5338314

2022-4-19

1 Introduction

Nowadays, screening techniques for cancer cells have received increasing attention and research. This project not only aims to ensure that tumours are detected as early as possible, but also to process and analyze the cell section medical imaging data in order to build an optimal multi-classification model for data prediction.

The problem addressed in this project is the multiclassification problem, with models constructed in a data prediction manner. The main challenge is that this is not a traditional dichotomous problem, but a quadruple classification problem. Since the data are normalized, the mean and standard deviation are calculated to observe the approximate level of dispersion. In addition, the cell slice image datasets were npy files, which we chose to pre-process by converting it to jpg format.

After the data was analysed, new challenges were identified. Due to the oversized images, random cropping was performed; again, due to the insufficient amount of data, data augmentation was used to perform feature spreading, data format transformation and data enhancement on the data. Several common classification learning algorithms were used to compare the effectiveness and accuracy of the models, including logistic regression, random forest, K-nearest neighbour, support vector machine, Gaussian Bayesian, decision tree classifier, AlexNet and ResNet.

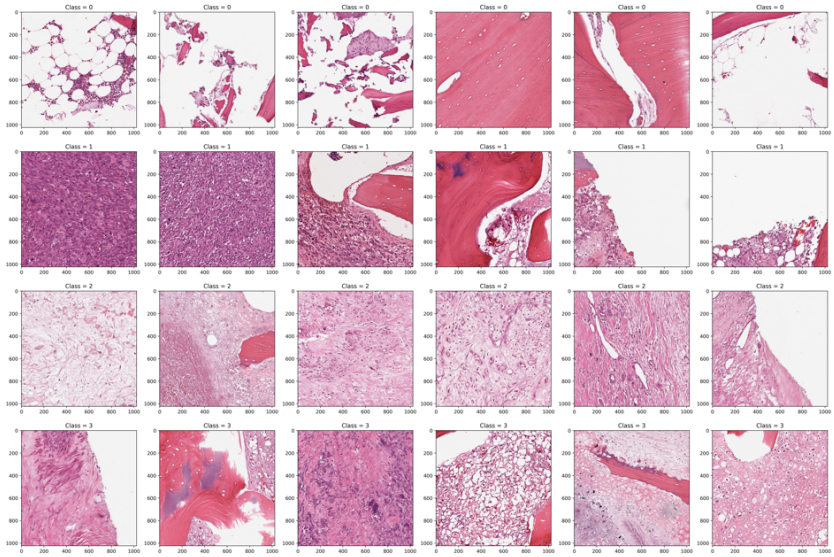
In order to optimize the model, the optimal parameters need to be found, namely the selection and tuning of the hyperparameters. Model selection was carried out based on accuracy and F1 scores, and several groups of models with the best results were selected, together with cross-validation and exhaustive enumeration to improve the confidence of the models. Separate predictions were made for X_test and the final classification was selected. Finally, our models achieved the desired accuracy and F1 score in the validation set.

2 Exploratory Data Analysis

When given a new dataset, it is important to understand the dataset, be familiar with the size of the dataset, see the statistical distribution of the data, and understand the correlation between features, etc. EDA can provide a better insight into the nature and cleanliness of the data, which can be a great reference for our data pre-processing, and also offer some guidance for our model selection.

2.1.1.Data Preview

These data are sections of human tissues, Class 0 indicates that no tumor is present, Classes 1-3 indicate that cancer is present, with each of these indicating a different type of tumor. With the naked eye, we can see that Class 0, healthy human tissue, shows a smooth, neat texture, while Classes 1-3 show an irregular, dense arrangement of invading cells. The similarity between class3 and class1 is high, and models can get confused. The image below shows a preview of the data from 4 different types randomly selected.



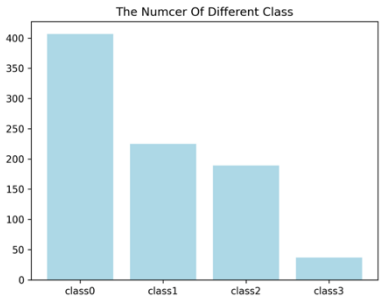
2.1.2.Statistical analysis of the sample

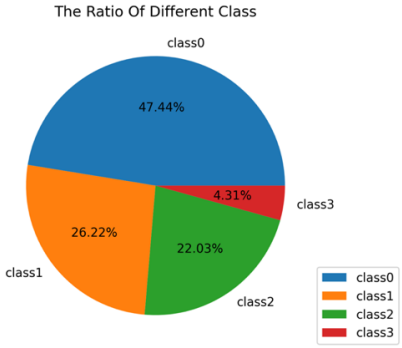
Since the data are normalized, we only calculate the mean and standard deviation to observe the approximate dispersion. From the table below, we can see that it is relatively easy to train the data with low dispersion. Moreover, the difference between X_train and X_test in data distribution is not significant, so in theory, the training set can provide good learning material for the model.

	X_TRAIN	Y_TRAIN	X_TEST
MEAN	0.76	0.83	0.77
STD	0.22	0.92	0.22
SHAPE	(858,1024,1024,3)	(858,)	(287,1024,1024,3)

2.1.3.Data class distribution

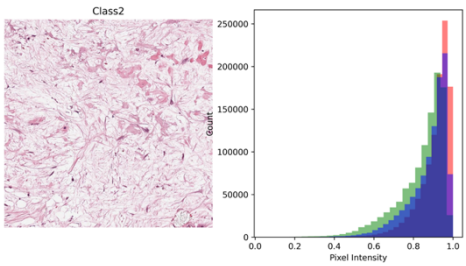
The class distribution of the training data set is shown in below. As can be seen, the distribution of different classes is uneven. Class0 makes up a large proportion and Class3 makes up a small proportion. Percentage of Class0 images is. 47% and Percentage of Class3 images is 4%.

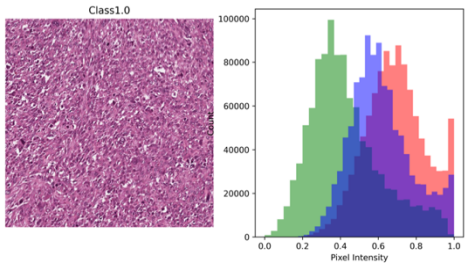




2.1.4.Three-channel analysis of color

Since our data is a 3-channel color image, this allows us to analyze the RGB value distribution of different Classes. It can be found that the RGB distributions of all class images basically overlap, which may lead to difficulties for algorithms like SVM/LR, which are better at data mining, to take advantage. For classification of such images convolutional neural networks will perform relatively better. This prediction is also confirmed during our experiments.



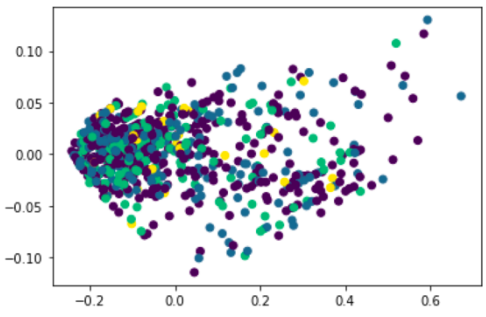


2.1.5.Data segmentation

Since we do not have y_test, we cannot use the original training set to train the models. To evaluate the effectiveness of the models, we use 20% of the data from the original x_train and y_train as the test set to evaluate the effectiveness of each model. The best model was predicted using the original training set and the test set for y_test.

2.1.6.PCA

In order to analyse the distribution of the data across all categories, Principal Component Analysis (PCA) was chosen to project the data into a two-dimensional plane for visualization. Based on the visualization results, it can be concluded that in the original distribution of the data, the differences between the different data in all categories are not significant. Therefore, in the method and model selection section, machine learning methods that offer more non-linear classification capabilities should be preferred, especially ensemble classifiers and deep learning methods.



3 Methodology

3.1.1.Data preprocessing

3.1.2.Data processing for machine learning models

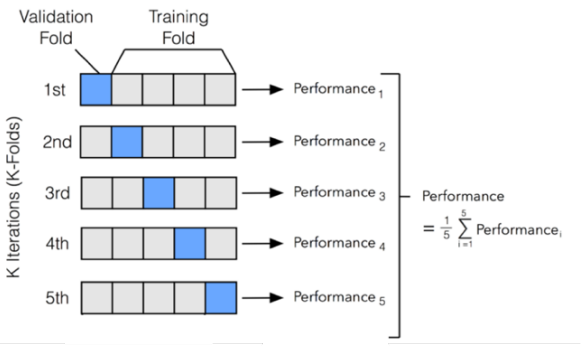
a) Random cropping

Based on the data exploration in Chapter 2, it is known that the data set is very large. Therefore, we take the approach of randomly clipping the data. This not only can reduce the memory consumption of the model, but also can improve the generalization ability of the model and prevent data overfitting. Theoretically the larger the image is the more features can be learned, but as the training data increases, the memory consumed by the model is gradually increasing. The following table shows the performance of different crop sizes, as the time increases, the power consumption increases almost exponentially, but the f1score does not improve much, so the computation time and f1_score are combined. 425 is chosen as the random crop size.

SIZE	F1-SCORE	ACC	TIME/S
224	0.36867	0.49419	201.88
324	0.39654	0.52326	434.26
425	0.40754	0.52907	731.37
525	0.39115	0.52326	1378.73
824	0.40256	0.52765	144473.78

b) K-fold Cross Validation

In this experiment, the training set is divided into k (k=10) subsamples, one single subsample is retained as the data of the validation model, and the other k-1 samples are used for training. Cross validation is repeated k times, once for each subsample, averaging k results or using some other combination, resulting in a single estimate. K-CV can effectively avoid the occurrence of over-learning and under-learning states, and the results are more persuasive[4]. K is taken as 10 because our data volume is less than small (858*20%), so a larger k value is needed for accuracy. Nevertheless, the larger the k is, the greater the consumption of the model. After considering the effect and consumption, we chose the more common k=10. (K-fold verification cross-validation, 2022)



3.1.3.Data processing for deep learning models

a) Data format conversion

For CNN networks, it has good image recognition capabilities. According to the pytouch tutorial, it is known that most of the functions of CNN networks are adapted to data in image formats such as .png/.jpg. Therefore, before entering res18 to start training, our group chose to convert the dataset in .npy format into images.

Since ResNet is a relatively deep network and our data volume is relatively small, we have copied 2 copies of each image when converting the images (X_train and y_train still keep the correspondence) so that we can get more training samples for training the model after the random cropping of data enhancement.

b) Data format conversion

Wang & Perez [1] (2017) mentioned that the performance of image classifiers can be improved by implementing appropriate image enhancements. In addition, overfitting on the model can also be reduced by image data enhancement. The generalization ability of the model can also be improved. In this project, we implemented a combination of different image enhancement methods, which include random cropping (parameter source pytouch tutorial), random level flipping, elastic transform and regularization. However, the data enhancement should not be excessive, because even if our training sample is expanded by a factor of two, the total amount is still small. Therefore, excessive transform operations will result in scattered data and the model will not fit well to the corresponding features.

TRANSFORM	ACC
NO DATA ENHANCEMENT	72%
RANDOM CROPPING, RANDOM HORIZONTAL FLIPPING	93%
RANDOM CROPPING, RANDOM HORIZONTAL FLIPPING, RANDOM CONTRAST, RANDOM BRIGHTNESS, ELASTIC TRANSFORMATION AND GRID DISTORTION	81%

3.2. Model selection

3.2.1. Machine learning based approach

3.2.1.1. SVM

The loss function used by SVM is the hinge loss function, and a study of the compatibility of alternative losses shows that when the proxy loss is a continuous convex function and is an upper bound on the 0-1 loss function at any value, the result obtained from solving the proxy loss minimization is also the solution to the 0-1 loss minimization. [2]. As the similarity between samples is described using a Kernel Matrix of the dataset, the number of elements of the matrix grows squarely with the size of the data. This makes the SVM calculation unmanageable as the size of the data increases.

3.2.1.2. Decision Tree

A decision tree is a predictive model that represents a mapping relationship between an object attribute and an

object value. It is a tree structure where each internal node represents a test on an attribute, each branch represents a test output, and each leaf node represents a category. Decision trees are also a very common method of classification.

3.2.1.3. Random Forest

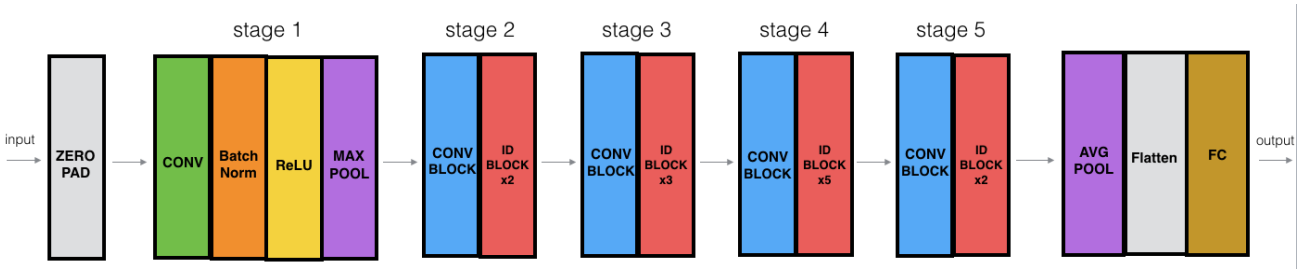
Random Forest, a classifier that uses multiple trees to train and predict samples, builds many different decisions tree classifiers on a subset of the original feature set and aggregates all the results, using the prediction that occurs most often. Random forests can handle very high dimensional (feature-laden) data without feature selection and are highly adaptable to datasets: they can handle both discrete and continuous data without normalization of the dataset; they are fast to train and can be used on large datasets; they are suitable for non-linear data due to their simplicity, high accuracy, and resistance to overfitting. The benchmark models.

3.2.2. CNN

Among several deep learning architectures, the convolutional neural network (CNN) is the most common one for image recognition tasks.CNNs consist of various convolutional layers with full connectivity. First, the image is given to a convolutional neural network, which passes through a series of convolutional, non-linear rectification functions (such as ReLU), maximal sets and fully connected layers, which then give the final output[3].

3.2.2.1. ResNet 18

The ResNet network solves the problem that deep CNN models are difficult to train. The VGG19 network has been modified and the degradation problem has been solved by adding residual units through a short-circuiting mechanism. The ResNet18 network has 17 convolutional layers and one fully connected layer. The structure of the residual network is roughly as shown:



3.2.2.2. AlexNet

The AlexNet model is an 8-layer structure, with the first 5 layers being convolutional and the next 3 being fully connected; there are 60 million parameters to learn and 650,000 neurons. Layers 2, 4 and 5 are all intra-GPU connections of the previous layer itself, layer 3 is fully connected to the previous two layers, and fully connected is 2

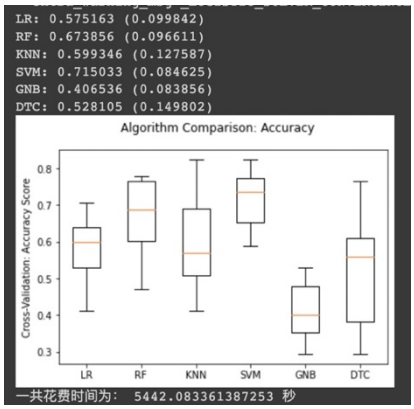
GPUs fully connected. reLU is after each convolutional layer as well as the fully connected layer.

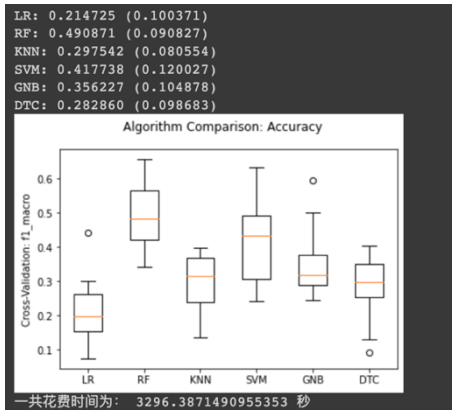
4 Result and Analysis

4.1. Comparing the performance of different algorithms

This step uses logistic regression, randomized forest, K-nearest neighbor, support vector machine, Gaussian plain Bayes, and eight algorithms such as decision tree classifier, Alex Net, and ResNet18. In which a 10-fold cross-validation is performed for each classifier algorithm to select the best model. Default parameters were used for all parameters[5].

The following figures show the accuracy (left picture) and f1score (right picture) of each of the 6 models after 10-fold cross-validation. considering both accuracy and f1score, we chose SVM, RF, and DTC for further exploration. Although Gaussian Parsimonious Bayes also performs well, it assumes a Gaussian distribution for the dataset, but obviously our data does not meet this criterion.





For CNN networks for it tends to have better results for images, one of the reasons is that it can retain the spatial information of each channel very well. And the neural network has a more complex computational process, so it has a much higher f1_score and acc compared to the classifier. We tried AlexNet, which was taught in class, and ResNet18, which has better performance. With the classical parameter design, lr=0,001, beath_size = 10, optimizer=SGD. decay LR by a factor of 0.1 every 10 epochs. they both have very good performance.

NET	F1_SCORE(MACRO)	ACC
ALEXNET	0.7240	88%
RESNET18	0.78	94%

Theoretically, the Res18 model would outperform AlexNet, but we have too little data, although after pre-processing we have tripled the training set. For such a mature neural network we do not have enough data to make a significant difference between them. At the meanwhile it is also possible that the model is able to capture their features quickly due to the large differences of our 3 classifications themselves.

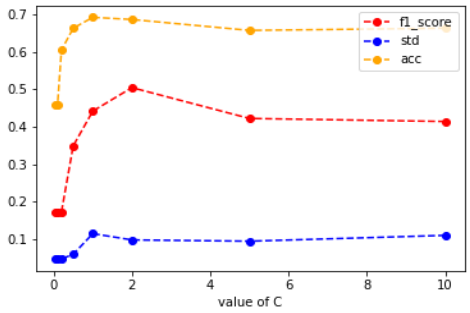
4.2. The Selection of Hyper reference

The adjustment process of Hyper-reference depends mainly on the f1_scour calculated by K-fold cross-validation, and accuracy as a reference term. We exhausted some common parameter values for the experiments.

4.2.1.SVM

The SVM model has two very important parameters, C and gamma, where C is the penalty factor, i.e., the tolerance for error. the higher the c, the less tolerant the error is and the easier to overfit. the smaller the c, the easier it is to underfit. the larger or smaller the c, the worse the generalization ability. The gamma implicitly determines the distribution of the data when mapped to the new feature space. Gamma in SVC in sklearn can take two values: 1. auto: 1 / n_features 2. scale: 1 / (n_features * X.std()).

For C we tested 12 values of C [10,7,5,3,2,1,0.5,0.2,0.1,0.05,0.001] and we can see that between 1 and 2 the f1score and acc of the model reach a maximum and after C>2 each indicator decreases. The table below shows the two different takes of gamma, and scale performs significantly better than auto.



	AUTO	SCALE
F1	0.372827	0.66042132
ACC	0.459302	0.6860465

4.2.2.Random Forest

The main parameters of the random forest are n_estimators (the number of subtrees), min_samples_leaf (the minimum number of samples of leaves), and min_samples_split (the minimum number of samples of branch nodes). We first tested the effect of the size of n_estimators on the model, shown below. The best is n_estimators=140.

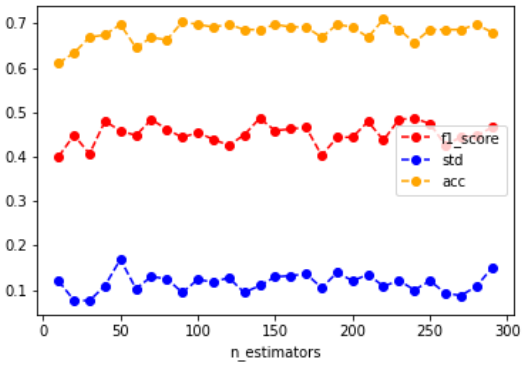
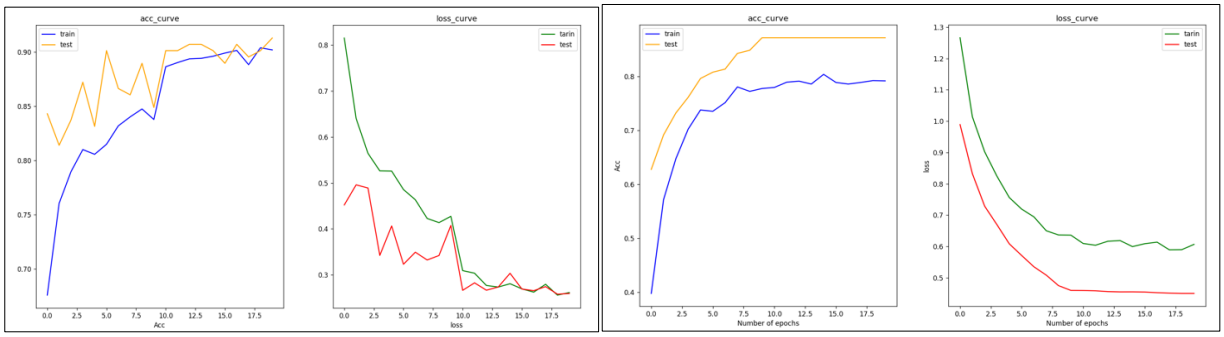


Table n shows part of the process of tuning the parameters for the random forest. The best performing parameters were n_estimators=140, min_samples_leaf=1, min_samples_split=3

N_ESTIMATORS	MIN_SAMPLES_LEAF	MIN_SAMPLES_SPLIT	F1_SCORE	ACC
140	1	2	0.6464	0.7209
140	2	2	0.6319	0.7093
140	3	2	0.6439	0.7034
140	1	3	0.6558	0.7035
140	1	4	0.6211	0.6977
140	1	5	0.6369	0.7034

4.2.3.AlexNet

We found that the default model parameters have very good performance (f1_score=0.82 acc=0.85). We tried different learning rates and found that the smaller the learning rate, the smoother the accuracy and loss curves of the model, as shown in the following figure.Fig.N lr=0.01 (left panel) and lr=0.001 (right panel) learning rate curves.

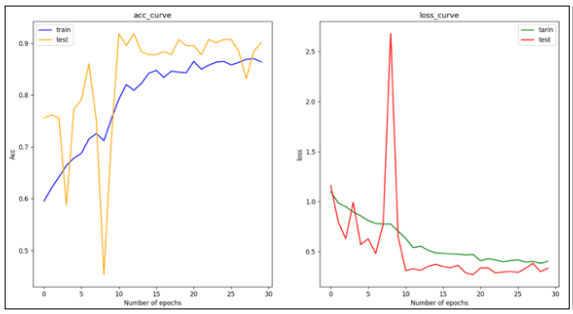


The learning rate curve in the above figure shows that the overall learning rate increases steadily at lr=0.0001 but does not change much after the 10th round. Therefore, we modified the parameters of the AlexNet coupon link layer Dropout to explore the performance of the model under different fitting states, as shown in the following table. Combining the above adjustments, we believe that the adaptation parameters for AlexNet are Dropout=0.2, lr=0.0001.

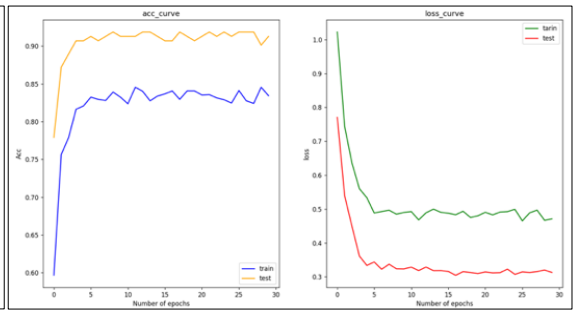
DROPOUT	F1_SCORE	ACC
0.1	0.7636	0.8837
0.2	0.7628	0.9170
0.3	0.7906	0.8779
0.4	0.8046	0.8502
0.5	0.8240	0.8537

4.2.4.ResNet 18

Although, AlexNet has performed well, for further exploration, we introduced a deeper network, ResNet18, for testing.Adam and SGDM are two of the best deep learning optimizers today. After experiments, we found that the accuracy and f1_score of Adma is slightly lower than those of SGDM at the same learning rate, although theoretically Adma is a better optimization algorithm. However, it may be due to the obvious classification of our data and the fact that the SGDM model is easier to find the global optimal solution for the image classification problem[6].



Performance curve of Adam



Performance curve of SGD

We then explore the learning rate, which determines the step size of the weight iterations and is therefore a very sensitive parameter that affects the model performance in two ways: the first is the size of the learning rate, and the second is the variation scheme of the learning rate[7].

LR	MOMENTUM	TRAIN LOSS	ACC	F1 SCORE
0.0001	0.9	0.3176	0.93698	0.92
0.001	0.7	0.3476	0.93024	0.87
0.001	0.8	0.3666	0.93123	0.88
0.001	0.9	0.4878	0.91867	0.88
0.005	0.9	0.6188	0.877907	0.66
0.01	0.9	0.6804	0.857558	0.63
0.05	0.9	0.7111	0.831395	0.63
0.1	0.9	0.8652	0.69	0.49

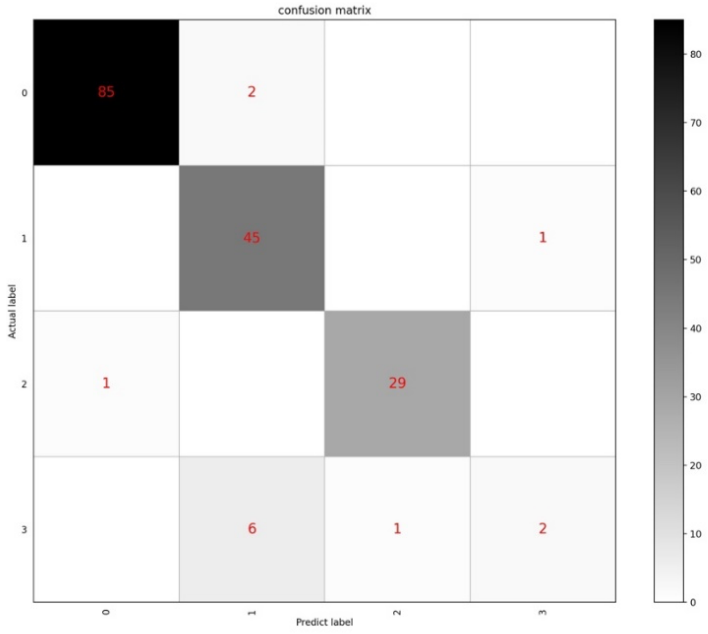
After adjusting the decay step of the LR, and finally found that the accuracy and F1score scores were the highest at step=10, 93% and 0.92, respectively. After fixing the step size to 10 and adjusting the gamma, we could find that the gamma did not have much influence on the overall experimental findings, which may be since the amount of our data is not much, and the epochs are small, the following table shows part of the process of tuning parameters.

STEP SIZE	GAMMA	TRAIN LOSS	ACC	F1 SCORE
10	0.1	0.3126	0.936035	0.92621
7	0.1	0.3133	0.938953	0.9071
6	0.1	0.3187	0.947674	0.9179
4	0.1	0.3750	0.927326	0.8871
3	0.1	0.4277	0.924419	0.8168
10	0.2	0.3844	0.924478	0.9128
10	0.5	0.3715	0.930243	0.8990

Most of the current deep learning models are optimized using the batch stochastic gradient descent algorithm. The batch_size affects the generalization performance, training time and stability of the model. The following table shows the performance of some batch_sizes. Finally, we choose beach_size=10

BATCH_SIZES	F1_SCORE	ACC
4	0.90656	0.90127
10	0.92621	0.93605
32	0.92508	0.94183

The final model we chose was ResNet18, with an optimizer of SGD, lr=0.0001, MOMENTUM=0.9, step_size=10, gamma=0.1, num_epochs=30, beach_size=10. The final performance of the model is F1_score=0.926218 and the accuracy = 0.9360465. The confusion matrix is shown in the figures below.



5 Discussion

5.1.1.Comparison of different models

According to our experiments, we found that SVM has the best performance for classifiers. However, the accuracy and f1 score of the convolutional neural network are significantly higher than those of the SVM. This is due to the fact that convolutional neural networks are already very good at image recognition, better at processing png and jpg images, and that CNN retains spatial information very well. CNN is a better choice for image classification because the process of feature extraction by convolutional operations results in an object being correctly identified in any region of the image.

5.1.2.Selection of different criteria

During the tuning process it was considered that each FEATURE was important. Although the four categories are unbalanced, the results are evenly distributed across the entire data set when the four categories are considered as having tumours versus not having tumours. In contrast, the distribution of the three classes representing getting the disease is uneven. Because the F1 score is a weighted average of precision and recall, the F1 score takes into account both false negatives and false positives, it is not as easy to understand as accuracy, but F1 is more applicable than accuracy, especially when the distribution of classes in the dataset is uneven. F1score = 2 ∗ Precision ∗ Recall / (Precision + Recall). Accuracy represents the ratio of the number of correctly predicted samples to the total number of

samples, Accuracy = TP + TN / (TP + TN + FP + FN). Therefore, in order to find the best combination of accuracy and recall in the case of an uneven distribution of categories, the F1 score was chosen as the primary metric and accuracy as a secondary metric, together with a confusion matrix to determine the model. Three algorithms were used to calculate the F1 score, average, weighted and macro. For a four-classification model, macro should theoretically give better performance, but the actual results we obtained were lower for macro than for weighted. The f1 score obtained with weighted was 0.92, whereas with macro the f1 score was 83. After some discussion, we decided that macro was the more appropriate algorithm. The reason for this is that although a higher f1 score represents a better model, it is necessary to prevent overfitting when the accuracy is too high. Therefore, considering the actual performance, the final choice was macro, under which we obtained an f1score of 0.83. While macro is lower than weighted, both perform well and f1 converges to a combined value, so we believe that the point found is more appropriate.

5.1.3.Future improvements

Although the data set has been expanded by doubling the data set and random filtering, a good result has been obtained. The amount of data in this project is still too small compared to the normal model and is not sufficient to achieve optimal results.

Another improvement is that we can make fuller use of the validation set to prevent overfitting. In this project, we only used k-fold cross-validation of the training set and evaluated the performance. It may be possible to find a more sophisticated and random way to select the training and test data, although this may cost more computational resources.

6 Conclusion

The aim of this project is to build a multi-label classifier. The input data is a sample of 1024*1024*3 three-channel RGB colour images, predicted to be four possible categories. After pre-processing the original data with random cropping, dataset segmentation, doubling, random filtering and image enhancement, our group used logistic regression, random forest, K-nearest neighbour, support vector machine, Gaussian plain Bayes, decision tree, AlexNet and ResNet respectively on the dataset. classifier, AlexNet and ResNet. After evaluation and comparison, among these algorithms, svm, logistic regression, ResNet and AlexNet all have a good accuracy rate, all above 70%. After fine-tuning the superparameter for each of these four models, the best performing model was ResNet 18 with SGD as the optimiser. While our f1score is calculated using two different methods, all comparative f1scores in this report are calculated in the same way. The F1 score for the final validation set and dataset was 0. 82 (macro) with an accuracy of 93%, indicating that the final model was sufficient to predict the categories on the unknown dataset.

7 Reference

[1] Wang, J., Pererz, L. 2017, 'The Effectiveness of Data Augmentation in Image Classification using DeepLearning', accessed 8 August 2019. < http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf>

[2] Zhang, T., 2004. Statistical behavior and consistency of classification methods based on convex risk minimization. Annals of Statistics, pp. 56-85.

[3] Shah, F.T. and K. Yousaf. Handwritten Digit Recognition Using Image Processing and Neural Networks. in World Congress on Engineering. 2007.

[4] Blog.csdn.net. 2022. K-fold verification cross-validation. [online] Available at: <https://blog.csdn.net/qq_36535820/article/details/119762665> [Accessed 16 April 2022].

[5] scikit-learn. 2022. 3.3. Metrics and scoring: quantifying the quality of predictions. [online] Available at: <https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score） > [Accessed 16 April 2022].

[6] Csdn.net. 2022. Adam Optimizer vs. SGD - CSDN. [online] Available at: <https://www.csdn.net/tags/OtDaUgzsNDUwODgtYmxvZwO0O0OO0O0O.html> [Accessed 16 April 2022].

[7] Blog.csdn.net. 2022. The impact of deep learning rate on model training. [online] Available at: <https://blog.csdn.net/qq_28531269/article/details/121108596> [Accessed 16 April 2022].

Code reeferenc

- https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html#saving-models
- https://scikit-learn.org/stable/auto_examples/cross_decomposition/plot_pcr_vs_pls.html#sphx-glr-auto-examples-cross-decomposition-plot-pcr-vs-pls-py
- https://www.bilibili.com/video/BV1s44y1i75r?spm_id_from=333.337.search-card.all.click