APPLIED MACHINE LEARNING SYSTEMS II (ELEC0135 22/23) REPORT

SN: 22039170

ABSTRACT

A malignant tumor in the brain is a life-threatening condition. The presence of Methylguanine methyltransferase (MGMT) promoter methylation has been shown to be a favorable prognostic factor and a strong predictor of responsiveness to chemotherapy. In this paper, machine learning based models are proposed to deal with the detection of MGMT promoter methylation instead of invasive surgeries taking brain tissues out of patients' body. Taking advantage of multiple modalities of MRI images, both Single Modality Model (SMM) and Multi-Modality Model (MMM) are proposed. The main building block of SMM is the feature extractor and that of MMM is feature extractor with the attention layer. By experiments and ablation study, it is found that multi-modal learning based model MMM performs better than SMM thanks to the attention mechanism.

Index Terms— MRI, multi-modal learning, attention mechanism, image classification, representation learning

1. INTRODUCTION

One of the most common malignant tumors, glioblastoma is usually found in adults. What makes it even worse, glioblastoma is also the one with the worst prognosis, with median survival duration less than a year.

A specific genetic sequence in the tumor known as Methylguanine methyltransferase (MGMT) is an essential gene encoding for a protein that can repair DNA[1]. Once it is methylated, it will transfer to inactivation mode which is detrimental to DNA repair progress including actively dividing one in malignant tumors. Korfiatis et al. suggest that radiation therapy and chemotherapy could be more effective to glioblastoma with MGMT promoter methylation [1]. Besides, MGMT promoter methylation has been shown to be a strong predictor of responsiveness to medication treatment like temozolomide [2]. Thus, it is very important to ascertain the existance of MGMT promoter methylation.

If traditional way of methylation analysis is chosen, surgeries will be carried out to the patient to extract a tissue sample of brain, which would be invasive and limited to availability of physical laboratory resources [3].

For numerous years, various medical imaging techniques such as Computed Tomography (CT scan), Positron Emission Tomography (PET), Magnetoencephalography (MEG) and Magnetic Resonance Imaging (MRI) have been employed to conduct brain tumor diagnosis in a computerized way. Essential clinical information including tumor presence, location and type could be inferred by imaging. MRI, a multi-modality imaging technique, is the most commonly used and effective method for detecting brain tumors due to its ability to distinguish between tissue and structure based on contrast levels. The method used for the competition to perform imaging of brain is MRI. Microscopic genetic changes (e.g. MGMT promoter methylation) may manifest as macroscopic morphological difference in the brain tumors that can be detected by MRI, which can serve as noninvasive biomarkers for determining methylation of MGMT regulatory regions.

Concretely, the result of imaging for brains with MGMT promoter methylation is different from ones without MGMT, which suggests that there might exist a certain causality between brain images and the presence of MGMT promoter methylation. The problem now becomes finding a model which could determine whether there is MGMT or not by the brain image of a subject. Therefore, one could take advantage of machine learning techniques to build and train such a classification model which could be later used to predict the presence of MGMT promoter methylation. Although sometimes training procedure may spend days if very large model is used, usually one day is totally enough for training binary classification model. In this way, invasive surgeries could be waived and huge amount of time could be saved, which means the patient will receive less invasive diagnoses and treatments. Then the possibility of survival of a patient might be increased with less physical damage and in-time correct therapy.

This paper proposes machine learning based method to detect the existence of MGMT in a brain. Concretely, there are MRI scans taken via 4 different methods: Fluid Attenuated Inversion Recovery (FLAIR), T1-weighted pre-contrast (T1w), T1-weighted post-contrast (T1Gd), T2-weighted (T2). Each scan is a binary image whatever the type of MRI scan it belongs to. It is necessary to first conduct data selection because there are several possible ways to utilize the given dataset and different machine learning model would be chosen if different strategy of data selection were to be applied. In this paper we propose two ways of utilizing the provided dataset. One is considering images for each MRI scan type separately as the input of proposed machine learning model. It is named as Single-Modality Model (SMM) in this paper. In this way there are four different possible input which is images of FLAIR, T1w, T1Gd, T2 separately. The other is using images from all these four MRI types as the input. It is named as Multi-Modality Model (MMM). Using this dataset building strategy, we could then build a multi-modal machine learning pipeline considering each of MRI

types as a single modality. The core building block of this model is the attention mechanism which can attend to features from each modality. The detailed description of models will be discussed in section three of this paper. The data preprocessing process and machine learning pipeline are implemented in python code which could be found https://github.com/N1ghtstalker2022/AMLS_II_assignment22_23. It could reproduce the experimental results described in this paper.

The rest of this paper is organized as follows. Section 2 reviews research work related to image classification and relevant application to MRI. Section 3 explains our proposed model and the rationale behind the choice. In Section 4, detailed description of dataset, preprocessing and implementation of machine learning models are included. In Section 5, several experiments are conducted to evaluate the effectiveness of the proposed model. Finally, in Section 6 we conclude the paper and provide future work.

2. LITERATURE SURVEY

Currently, analysis of MRI is conducted manually most of the time and it could spend much time for clinicians to locate and segment tumors for later treatment [4]. In order to resolve these issues, researchers have started finding feasible machine learning based methods to tumor detection and segmentation.

Taking advantage of support vector machine (SVM), Korfiatis et al. obtain the features of the MRI by handcrafting a feature detection filter and using a sliding window manner [1]. SVM is an interpretable machine learning classification algorithm with clear decision boundary. It could be trained effectively even with relatively small dataset like one pertain to medical images.

On the other hand, it is often non-trivial to construct filters by hand to capture effective features for MRI images. Recent years, researchers have tried out varieties of deep learning based method to recognize possible patterns within MRI images and classify them.

G. Raut et al. argues that MRI can generate images with patterns which may not be visible to the naked eye but can be percepted by machines and they proposed a machine learning model using convolutional neural network (CNN) to determine if there is a tumor in a given image [5]. CNN is basically a neural network architecture with the assistance of convolution operation and pooling operation. By apply convolution to a target image, it could extract features as hand crafted filters used in traditional image processing could detect edges [6]. Unlike hand crafted convolution filters, the weight of convolution matrix in CNN is not fixed but could be trained through back-propagation algorithm.

He et al. proposed ResNet which mitigate the problem of degradation problem (the performance of the model drops when the network becomes deeper and deeper) [7]. The contribution of ResNet whose building blocks are still mainly CNN is that it introduces identity mapping (also known as skip connection). It alleviates the notorious gradient vanishing problem, which allows researchers to train deeper neural networks and gain better model performance. It turns out that by introducing residual learning and identity mapping, ResNet performs better than plain CNN networks on image classification problems[7]. It is also proved that ResNet could generate decent results on MGMT status classification [8].

Convolutional Recurrent Neural Network (CRNN) proposed by Shi et al. combines CNN and Recurrent Neural Network (RNN) to an integral machine learning model [9]. Basically, it could extract features within a given image by CNN and model spatial dependencies of learned features. Han and Kamdar present the use of CRNN on MRI images to recognize the state of MGMT promoter methylation [2]. Since one MRI scan generates a series of frames of brain, in their study, each frame of an MRI scan is treated as single input to CNN. Once CNN extracts feature for all frames of a MRI scan, those features are feed into latter bidirectional RNN to analyze its spatial dependency and output a predicted class.

Tan and Le proposed EfficientNet achieving better accuracy than previous ConvNets but with an order of magnitude fewer parameters. It uses MBConv as main building blocks. The key idea for EfficientNet being efficient is its novel compound scaling method which uniformly scales network width, depth and resolution with a compound coefficient. In their work, eight EfficientNet sub-models (EfficentNet-B0, EfficentNet-B1 to EfficentNet-B7, whose corresponding required FLOPs progressively increase) were designed and tested. In the study of Raut et al., they found EfficientNet-B0 performs better than former CNN architectures like AlexNet and ResNet [5]. They also demonstrated how fine-tuning on pretrained EfficientNet could obtain better results compared with pure transfer learning without fine-tunning.

When tackling problems including image captioning, visual question answering, video action recognition, and audio-visual speech recognition, researchers have found out that models utilizing multiple modalities outperform ones selecting single modality. Provided the intuition that models aggregating multiple modalities perform better than their single modality counterparts since more information is included by multiple modalities, it is further proved theoretically that multimodal learning generally performs better than just using single model [10].

Recently, with the success of deep learning, some researchers have combined multimodal learning with deep learning techniques, achieving empirically remarkable results in segmentation tasks with MRI multimodality images [11, 12]. In addition, multimodal fusion is the essential approach in multimodal learning to combine information from multiple modalities. Qu and Xiao proposed an attentive multi-model CNN for Tumor classification based on MRI by involving attention mechanism for feature fusion [13]. In their work, they also demonstrate that their work outperforms the state-of-the-art (SOTA).

Attention mechanism was first introduced by Bahdanau et.al to an encoder-decoder architecture for neural machine translation task [14]. It allows the model to automatically focus on the relevant part of the source sentence when translating a next word. Vaswani et al. further described attention mechanism as given a set of query, keys and values computing the weighted sum of values where the weights are computed by a certain scoring function of the current query with keys [15]. They also propose the notable attention-based Transformer model which has become the foundation for state-of-the-art natural language processing models like OpenAI's GPT-series.

3. DESCRIPTION OF MODELS

By utilizing provided MRI scans, the goal of this project could be described as train a machine learning model to take input MRI images and classify them into categories (brain tissues with or without MGMT promoter methylation).

3.1. Single modality model

If only one of provided four modalities of MRI scans is considered, the task degrades to the category of binary image classification problem.

In this project, several machine learning models including Plain Convolutional Neural Network (PCNN), ResNet-34, EfficientNet-B0 are applied. After experimentations, we compare these models using selected evaluation metrics.

PCNN

PCNN, also known as CNN, is apt at extracting high level features for input images (or matrices). As shown in Fig. 1, the building block of a PCNN is convolutional blocks which contains convolutional layer, activation layer and pooling layer. The number of convolutional blocks for a PCNN could be from several to more than dozens.

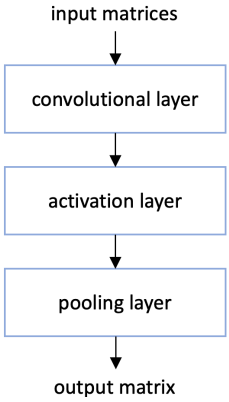

Fig. 1 The structure of a convolutional block.

The convolutional layer performs the convolution operation meaning applying a certain number of filters to the input matrices. The convolution operation could be mathematically formulated as in Eq. 1.

$$O(i,j,k) = \sum_m \sum_n \sum_c X(i+m, j+n, c) * W(m,n,c,k) + b(k) \quad \textbf{Eq. 1}$$

Where X denotes 3D input data (with i and j representing location and c representing channel), W denotes 4D convolutional filters (with m and n representing location, c representing input channel and k representing filter number), b denotes the bias term and O denotes the output tensor after convolutional operation. The number of output channel is exactly the same as the number of filters. Unlike traditional edge detection filters like Sobel filter, the filters involved in convolutional layer are learnable. In the training process, the parameters in filters could be changed by learning algorithms like gradient descent. Each filter for a convolutional layer could learn different features from any other filters. Fig. 2 shows the operation in convolution layer.

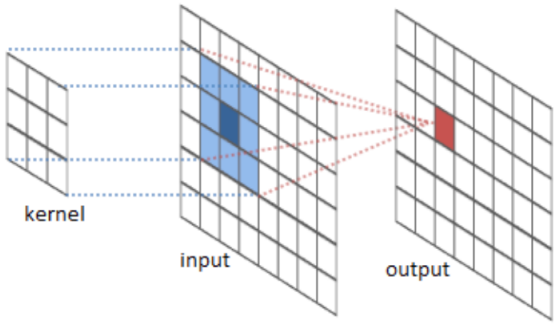

Fig. 2 The convolution operation. A parametrized kernel (filter) multiplies with the input matrix in an elementwise way.

The activation layer can introduce nonlinearity to feature maps obtained from the previous convolutional layer. The key effect of it is allowing the model to simulate complex functions by passing data through the activation function. Recently, one of the most widely used activation function in neural networks is Rectified Linear Unit (ReLU). Both its original form and derivative is simple and computationally efficient, which facilitate the backpropagation during training. It also alleviates vanishing gradient problem by its non-saturating feature, which means that its derivative is either zero or one.

$$ReLU(x) = max(0, x) \quad \textbf{Eq. 2}$$

$$\frac{d}{dx} ReLU(x) = \begin{cases} 0 & if\ x < 0 \\ 1 & otherwise \end{cases} \quad \textbf{Eq. 3}$$

The pooling layer downsamples the size (height and width) of the feature maps produced by previous convolutional and activation layers. By downsampling, it reduces the number of parameters while preserving parameters which could

represent their local feature maps to some extent. Additionally, by involving pooling operation, the impact exerted by micro fluctuations of parameters is alleviated, which means that the possibility of overfitting drops and the model is more robust.

On account of its relatively simplicity and effectiveness, PCNN is selected as a baseline model for the MRI image classification task to determine the existence of MGMT promoter methylation.

ResNet

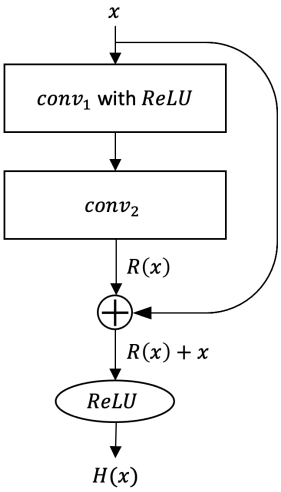

Fig. 3 Residual block used in ResNet. $\boldsymbol{conv_1}$ and $\boldsymbol{conv_2}$ represent two convolutional layers respectively.

ResNet is also used in this project. The core idea of ResNet is reformulate pure learning layers to ones with reference to the input of layers. ResNet involves residual learning by introducing identity mapping, which alleviate the degradation problem (with the network depth increasing, the model performance saturates and then degrade quickly).

Consider $H(x)$ as underlying mapping generated by plain neural network blocks and identity matrix which is the original input to the network block. It is assumed that if a complicated function like $H(x)$ could be asymptotically approximated by nonlinear layers like neural networks its corresponding residual function (i.e., $H(x) - x$) could also be asymptotically approximated [16]. Therefore, rather than let neural network approximate $H(x)$, skip connection is designed to let neural network conduct approximation to the residual function $R(x)$ where $R(x) = H(x) - x$. The function passing through the residual block then could be represented as $H(x) = R(x) + x$.

Particularly, in the context of deep residual learning, $R(x)$ could be formulated in a more specific expression:

$$R(x) = conv_2\left(ReLU\left(conv_1(x)\right)\right) \qquad \textbf{Eq. 4}$$

where $conv_1$ and $conv_2$ denotes the convolutional layer as demonstrated in Fig. 3. The operation $R(x) + x$ is performed by a shortcut connection and elementwise addition. Then another $ReLU$ is appended after the addition, which generates the final output of a residual block. In this project, ResNet-34 is used, which is constructed by one convolutional layer, one max pooling layer, a series of residual blocks followed by an average pooling layer and finally one fully connected layer. It follows the overall architecture demonstrated in Fig. 4. The detailed implementation of it will be illustrated in the later section 4.

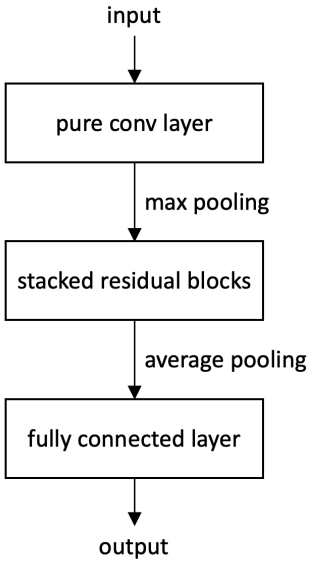

Fig. 4 The overall architecture of ResNet.

EfficientNet

The core contribution of EfficientNet is introducing an innovational compound scaling method. For a specific deep learning based model, there are usually three factors that could have an influence on the model performance. They are network width, depth and image resolution. Instead of scaling any one of those factors which rapidly lead to accuracy saturation for bigger models, the compound scaling method uses a compound coefficient φ to uniformly scale network width, depth and resolution denoted as w, d, and r respectively.

$$\begin{aligned} w &= \beta^{\varphi} \\ d &= \alpha^{\varphi} \\ r &= \gamma^{\varphi} \\ s.t.\ \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\ \alpha \geq 1, \beta &\geq 1, \gamma \geq 1 \end{aligned} \qquad \textbf{Eq. 5}$$

Where α, β, γ represent weights assigned to width, depth and resolution respectively whose values are determined by grid search algorithm. Intuitively, φ is a user-specified coefficient that controls the amount of available computing resources. Since the floating point operations (FLOPs) of a convolution operation is proportional to w, d, and r, scaling a convolutional block with Eq. 5 will roughly rise total FLOPS up with a multiplicative factor $(\alpha \cdot \beta^2 \cdot \gamma^2)^{\varphi}$. The architecture of EfficientNet mainly consists of plain convolution blocks and

mobile inverted bottleneck (MBConv). The baseline model for EfficientNet is named as EfficientNet-B0 whose scaling factors α, β, γ, are found under the optimization criterion $ACC(m) \times \left(\frac{FLOPs(m)}{T}\right)^{w}$ Where m denotes current model, ACC(m) and FLOPs(m) represent the evaluation accuracy and FLOPs of the model m respectively, T is the target FLOPs and w is a hyperparameter controlling the trade-off between accuracy and FLOPs. After finding optimal α, β, γ, we keep them as constants and increase the compound coefficient φ to obtain EfficentNet-B1 to B7.

3.2. Multimodality model

In order to fully take advantage of four modalities of the MRI scans, the multimodality model (MMM) is proposed. We use the same feature extractors for SMM mentioned in section 3.1 like CNN and ResNet to learn features for each modality.

Since each modality of MRI focuses on different aspects, it might be easier to detect macroscopic morphological difference (the status of MGMT promoter methylation) in the brain tumors by using some modalities but more difficult by others. In another word, it would be beneficial to assign higher weights to features which could be more useful for MGMT promoter methylation detection.

In this case, attention mechanism is applied to selectively concentrate on more relevant features among those learned from modalities including FLAIR, T1w, T1Gd and T2w. Theoretically, by selectively focusing on important features from each modality, attention mechanism can help to improve the accuracy and robustness of multi-modal classification models.

The inspiration of attention mechanism could be originated from hash table where keys used to query values in the table are converted to the de facto index by the hash function.

Let α be a one-hot encoding of the index of hash table. The operation of lookup could be mathematically formularized as $\alpha \times V$.

For attention mechanism however, the idea is to retrieve a soft convex combination over all values instead of a hard assignment to one specific value. It generalizes the hash function to a score function (Eq. 6) between a query Q and the set of keys K.

$$z = score(Q, K) \qquad \textbf{Eq. 6}$$

Then, softmax function is applied to realize convex combination which requires the sum of weights equal to one.

$$softmax(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)} \qquad \textbf{Eq. 7}$$
$$\text{for } i = 1, \dots, K \text{ and } z = (z_1, \dots, z_K) \in \mathbb{R}^K$$

Then we could obtain the attention weights which could be expressed as:

$$\alpha = softmax\big(score(Q, K)\big) \qquad \textbf{Eq. 8}$$

Where α represents attention weights. One could further multiply attention weights to the set of values V to obtain the output which could be fed into downstream classifier. The final output of attention could be formularized by Eq. 9:

$$Attention(Q, K, V) = softmax\big(score(Q, K)\big) \times V \qquad \textbf{Eq. 9}$$

Based on the attention mechanism, we could let the model attend to modalities which may have more impact on the classification result. However, attention weights generated by a single query and the set of keys may not be capable of attending useful features of values comprehensively. Therefore, rather than using a single query, we use multiple queries which presumably could allow the model to jointly attend to multiple subspaces of features from different modalities. Eq. 10 illustrates the key idea of multi-head attention.

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$
$$\text{where } head_i = Attention\big(QW_i^Q, KW_i^K, VW_i^V\big) \qquad \textbf{Eq. 10}$$

W_i^Q, W_i^K, W_i^V are learnable linear transformations for Queries, Keys and Values respectively. The output of each attention head is concatenated and multiplied by a transformation matrix W^O to obtain the final output of multi-head attention.

The overall MMM structure is shown in Fig. 5.

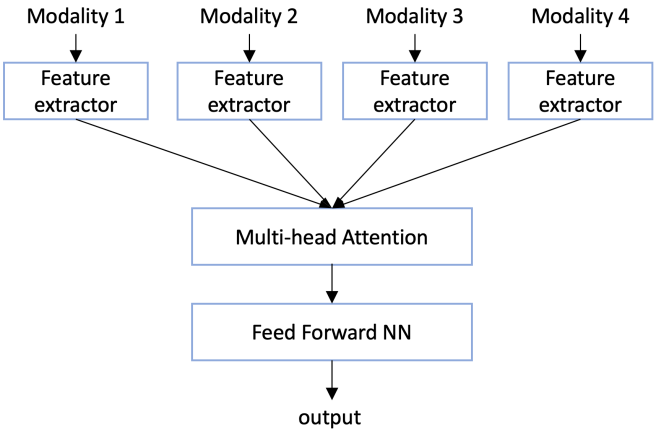

Fig. 5 The overall structure of Multi-modality Model.

4. IMPLEMENTATION

4.1. Dataset Description

The dataset used in this paper is a multi-center brain tumor MRI dataset called RSNA-MICCAI dataset [17].

The original dataset contains MRI testing results for 585 patients, including both four modalities of the MRI scans and

the existence of MGMT label. Concretely, in original 'train' dataset there are 585 folders of which each one is denoted by five-digit identification number. Each sample folder consists of four sub-folders corresponding to the four modalities of the MRI scans, including fluid attenuated inversion recovery (FLAIR), T1-weighted pre-contrast (T1w), T1-weighted post- contrast (T1Gd), and T2-weighted (T2), obtained from the video cut frames acquired by imaging. The imaging process is focused on specific aspects for each modality or type of scan. For instance, FLAIR emphasizes other parts by suppressing liquid signals like water after cerebrospinal fluid (CSF) suppression. T2-weighted, however, emphasizes the distinction in lateral tissue relaxation and comprehensively depicts of the lesion from various viewpoints is achieved by combining different effects. A quadruple of the four distinct imaging modalities constructs each sample in the dataset. For each modality of a given subject, the number of images is different, as it is shown in Table. 1.

Table. 1 Number of images for each scan type of a given subject. Here five examples of subject are provided.

Subject	FLAIR	T1w	T1Gd	T2w	MGMT
00000	400	33	129	408	1
00002	129	31	129	384	1
00003	129	33	129	408	0
00005	400	28	129	424	1
00006	129	32	129	408	1

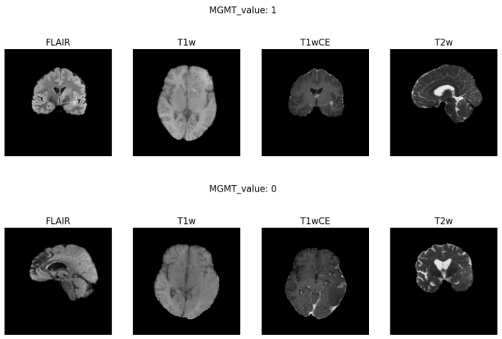

Fig. 6 Two samples of MRI scans composed of four modalities. Images in the first row represent FLAIR, T1w, T1Gd, and T2 modalities of a positive sample, and those in the second row represent those modalities of a negative sample.

Fig. 6 shows four modalities of a positive sample where value of MGMT is denoted as one and a negative sample where value of MGMT is denoted as zero. The image for each modality is selected in the middle of the frames. Since the images for a given modality of a subject are obtained from video frames, it is necessary to explore the patterns of those frames. We visualize some of those frames by using pydicom package in python. Fig. 7 is one example from which we could speculate that images in the middle of the sequence contains more information because their average pixel intensities are higher than the rest.

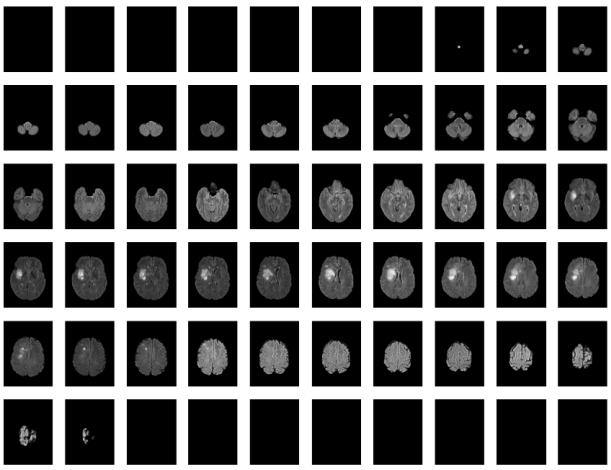

Fig. 7 Sixty sequential images a subject.

4.2. Data preprocessing

As it is demonstrated in Fig. 8, for each modality the number of DCM file of different subjects varies and ranges from 15 to as large as 514, but some values for each scan category are over-represented. Given limited computing resources, it is necessary to select effective part from the whole dataset.

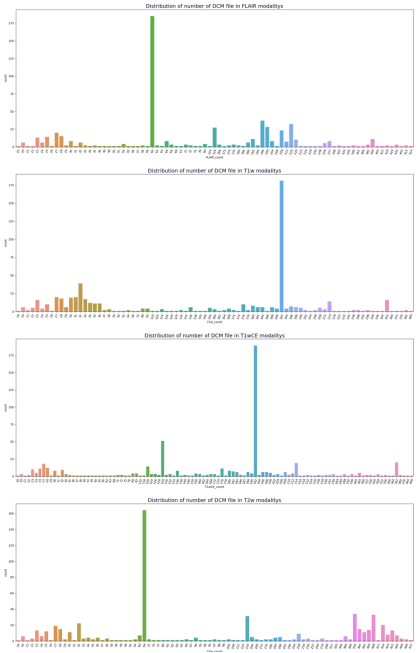

Fig. 8 The distribution of number of images for FLAIR, T1w, T1Gd, and T2 modalities. The image is stored in the form of DCM file.

It is noticed that a certain number of images at the beginning or at the end of the sequence has a lot of black area (e.g., Fig. 7). These images will therefore be useless in our models and may even cause over-training. When creating the

image sequences, we will therefore start from the central image of each folder and we will then take the same number of images upstream and downstream.

In this paper, two data selection strategies are implemented and experimented. The first one is to choose exactly one image located in the middle of the sequence of frames. As it is analyzed from the sequence images, the image in the middle usually has more valid pixels (the ones with intensity larger than zero). The other is to choose 10 images in the middle of a sequence of frames. Each one of these 10 images could be treated as a single data point with the same label corresponding to the subject. In this way, there could be ten times more data than the first data selection strategy.

It is note-worthy that the size of original images in the dataset varies. Since the machine learning model used in this paper is deep learning-based model which requires the same dimension of the input, it is indispensable to uniform the image size. Also, image crop, rotation and flips are carried out to introduce more variability and diversity in the training data, which could benefit the machine learning model in terms of its generalization capability. In addition, image normalization is implemented by bring the pixel values of a given image to a common range (e.g., from zero to 1). It could help improve the convergence of machine learning algorithms during training by reducing the variance of the input. In the code, we took advantage of existing functions from torchvision transformation to conduct image preprocessing.

In actual implementation, it is essential to split dataset into training, validation, and test set for which different image transformation steps are applied. Concretely, train_test_split() function from sklearn.model_selection is used to first split the whole dataset into training and testing dataset. Then in this paper, the training dataset is further used to perform k-fold cross validation. Stratified K-Folds cross-validator called StratifiedKFold from sklearn.model_selection provides train/validation indices to split data in train/validation sets while preserving the percentage of samples for each class. Particularly, 4-fold cross validation is conducted and for each fold the train set generated from the first step is split into training and validation set with a ratio of 3:1, which leads to the final ratio of 3:1:1 w.r.t training, validation and test set. In addition, stratification ensures each fold holding a similar distribution of the target label. By stratified k-fold cross validation, the data is partitioned into k subsets. Then one of k subsets is selected to perform validation and the rest k-1 subsets used for training, which is repeated k times. The performance of the model is calculated by averaging the metric generated by k iterations. This could reduce the risk of overfitting and provide a more reliable estimate of model performance. Fig. 9 Shows the overall strategy for data split.

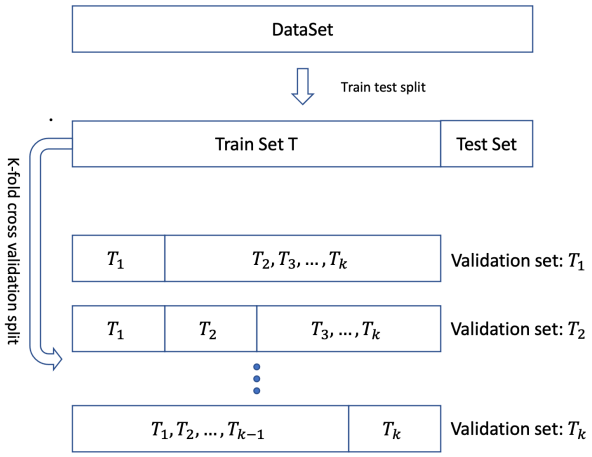

Fig. 9 The process of data split. First the whole dataset is split into train and test set. Train set is denoted by T. Then by k-fold split, the train set is further divided into k folds where each fold is composed of validation set $\boldsymbol{T_n}$ and train set $\boldsymbol{T_1, \dots T_{n-1}, T_{n+1}, T_k}$ $\boldsymbol{(n = 1, 2, \dots, k)}$.

After splitting the data, necessary image transformation is conducted for training, validation and test set separately. For training set, images are cropped with ratio between 0.8 to 1.2 and resized to dimension 224x224 by RandomSizeCrop() function. Then RandomRotation() function is applied to carry out image rotation with its degree between 0 and 360. RandomHorizontalFlip() function is used to perform image flip horizontally. Finally, z-score normalization is applied by Normalize() function with mean and standard deviation calculated from training data. The z-score formula shown below in equation (11) subtracts the mean value of the channel from each element of the channel and then divides by the standard deviation of the channel, resulting in a normalized output tensor with a mean of 0 and a standard deviation of 1. Since all images are binary, there is only one channel in this case.

$$normalized\_value = \frac{original\_value - mean}{standard\ variation} \qquad (11)$$

However, for validation and test set, image resizing which is the only preprocessing conducted is implemented by Resize() function from torchvision transformation package, because when it is more reasonable to use original image pattern (no rotation or flipping) to evaluating the model performance.

4.3. Machine Learning model implementation

We create our customed dataset BrainImageDataSet extending Pytorch standard DataSet class from module torch.utils.data. Then DataLoader class in the same module is used to load data when training, validation and testing.

Three key neural network based building blocks are PCNN, ResNet-34 and EfficientNet. PCNN and ResNet-34 are implemented from scratch using torch.nn module in pytorch,

while the EfficientNet-B0 model is imported directly from efficientnet_pytorch module.

For our Multimodal model (MMM), the key difference is adding a Multi-Head Attention stage. Concretely, we use the same feature extractors as for SMM, instead of feeding the extracted features directly into the classifier, we apply multi-head attention to those features and after that we pass the attention output to the classifier. We use the same variety of feature extractor to all modalities but the feature extractor for every modality does not share model weights in order to learn its own subspace of features. In python implementation, MultiheadAttention class in PyTorch torch.nn module is used to perform multi-head attention mechanism.

As mentioned in previous section 4.2, k-fold cross validation strategy is applied to the model training and validation. In this case, one fold is reserved for validation and the rest is used for training.

The operation of feeding the whole dataset to the machine learning model is called an epoch. In the experiment, the number of epochs is set to 20. During an epoch, the model is trained on all the samples in the dataset for a certain number of iterations which depends on the training batch size. The calculation of number of iterations could be expressed as $n_{iter} = \left\lceil \frac{n_{sample}}{batch\_size} \right\rceil$, where n_{iter} denotes number of iterations, n_{sample} denotes the number of total samples of dataset.

For each iteration, the optimizer updates the model parameters based on back propagation.

Specifically, Adaptive Moment Estimation (Adam) optimizer is used to adapt the learning rate of each weight based on the average of the magnitudes of the gradients and the second moments of the gradients. In our python implementation, we use torch.optim.Adam class from PyTorch library with a learning rate of 0.001. To update the model's parameters during each iteration of the optimization algorithm, it is important to set reasonable batch size which is the number of data point. The batch size for training and validation process is set to 64. If batch size for training is too small, it can lead to more noise in gradient updates and hence slower convergence and possible stuck in poor local minima. If that is too large, the model may convergence very fast but it could suffer overfitting problem and memory constraints of computer hardware. It is also very important to define a loss function to calculate the difference between the model output and real value. We use the cross entropy loss in PyTorch torch.nn module. Since we use batch gradient descent strategy for training, cross entropy loss could calculate the difference between distribution of output values and real values for 64 samples in a given batch.

Early stopping is also implemented in training pipeline to prevent overfitting problem. During the training loop, if the validation loss has not improved for ten consecutive epochs, early stopping will be triggered, and the training loop will be terminated.

With selected hyper-parameters and developed training pipeline, we train and validate our proposed model SMM and MMM with our preprocessed dataset. We made a comparison among learning curves of models using PCNN as feature extractor, where the result is plotted in Fig. 10. It suggests that the model does not converge very well within twenty epochs especially when there is less training sample using data selection strategy one. The reason for model underfitting could be the naïve structure of the feature extractor PCNN. The model could not effectively learn complicated features for MRI image with a simple feature extractor.

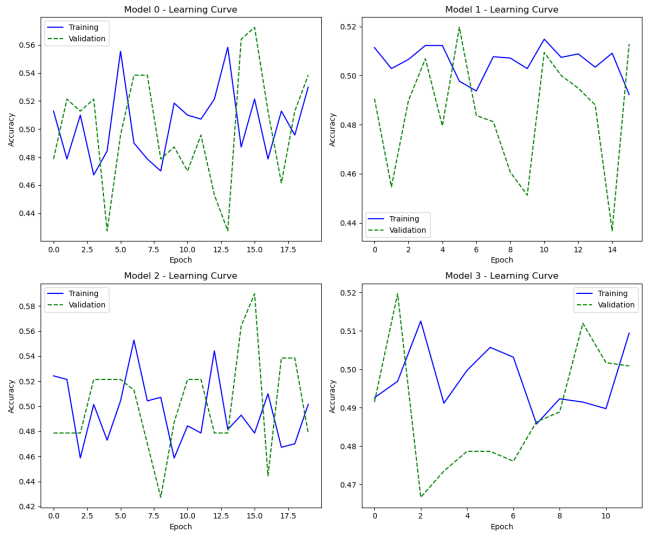

Fig. 10 Learning curves for models. Model 0 represents SMM using PCNN with data selection strategy one. Model 1 represents SMM using PCNN with data selection strategy two. Model 2 represents MMM using PCNN + Attention with data selection strategy one. Model 3 represents MMM using PCNN +Attention with data selection strategy two. The data selection strategies are explained in section 4.2.

We also made an ablation study towards the effectiveness of feature extractors and multi-head attention. The hyper-parameters remain the same as before. As in Fig. 11 shows, model MMM with PCNN + attention and MMM with ResNet-34 + attention have better validation accuracy than others. The early stopping criterion for all three MMM model does not work, which means with more epochs the model could converge better.

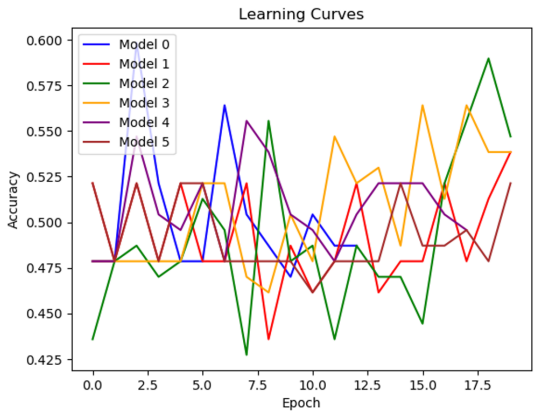

Fig. 11 Learning curve for ablation study. Model 0, 1, 2, 3, 4, 5 represents SMM with PCNN, SMM with ResNet-34, SMM with EfficientNet-B0, MMM with PCNN + attention, MMM with ResNet-34 + attention, MMM with EfficientNet-B0 + attention, respectively.

5. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present the experimental results and analysis of our proposed method SMM and MMM. We conducted a series of experiments on test set with data selection strategy one to evaluate and compare the performance of our model. Concretely, we define three feature extractors for both SMM and MMM: PCNN, ResNet-34, EfficientNet-B0. Then we train these six models separately and test them with test set. The evaluation metrics used in our experiments include accuracy, precision, recall, and F1-score. As it is shown in Table. 2, MMM generally perform slightly better than SMM under this evaluation criterion, from which we could infer that changing feature extractors to more complicated ones seems to have non-trivial effect on model performance.

In addition, it is argued that the key factor that enhances the model performance could be the attention layers applied in MMM, since attention mechanism allows the model focuses on features from modalities that could be more useful for MGMT status classification.

Table. 2 Displays the testing result for all six models. In the table, SMM_0, SMM_1, SMM_2, MMM_0, MMM_1 and MMM_2 denote SMM with PCNN, SMM with ResNet-34, SMM with EfficientNet-B0, MMM with PCNN + attention, MMM with ResNet-34 + attention, MMM with EfficientNet-B0 + attention.

Model	Accuracy	Precision	Recall	F1-score
SMM_0	0.48	0.47	0.48	0.46
SMM_1	0.39	0.39	0.39	0.38
SMM_2	0.54	0.54	0.54	0.54
MMM_0	0.56	0.57	0.56	0.54
MMM_1	0.55	0.55	0.55	0.55
MMM_2	0.48	0.23	0.48	0.31

We also draw the confusion matrices for aforementioned six models respectively in Fig. 12. It is suggested that the model generally predicts better when there is no MGMT promoter methylation.

However, the performance of the model is not desirable. It is very likely that a patient with MGMT promoter methylation will be classified as the opposite status. We argue that it is because the models could be still underfitting since there is too little training samples in data selection strategy one. It is difficult for our deep learning based models to learn complicated features using relatively small dataset.

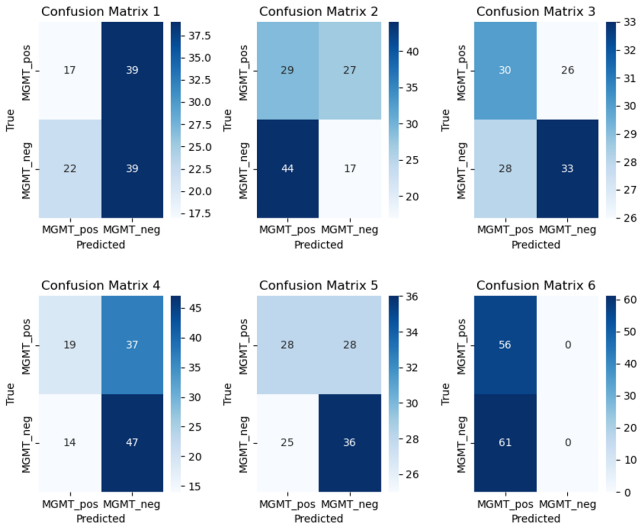

Fig. 12 Confusion matrices for aforementioned models in Table. 2: SMM with PCNN, SMM with ResNet-34, SMM with EfficientNet-B0, MMM with PCNN + attention, MMM with ResNet-34 + attention, MMM with EfficientNet-B0 + attention respectively. MGMT_pos represents there is MGMT promoter methylation in MRI image while MGMT_nege represents there isn't MGMT promoter methylation.

6. CONCLUSION

In this paper, we propose single modality model (SMM) and multimodality model (MMM) to find macroscopic differences between MRI images with MGMT promoter methylation and ones without MGMT promoter methylation. We design two data selection strategy with the given dataset with four modalities and conduct data augmentation by cropping, resizing and normalization. We train, validate both SMM and MMM with different feature extractors and test the trained model on test set. Since we have limited computing resources, we train models with ResNet-34 and EfficientNet-B0 as feature extractors under data selection strategy one which could generate less data sample. It turns out that MMM model generally perform better than SMM model thanks to multimodal learning with attention. However, the

performance improvement by training with multimodalities is relatively small. We argue that one possible reason could be the insufficient training samples. If GPU or TPU resources are available to use, we could feed a lot more data samples to our models to learn features more effectively. Another possible reason is that we neglect the relationship for images of one modality of one patient. Therefore, in future work, we could improve our network architecture by making feature extractor capable of capturing in-frame information.

7.REFERENCES

[1] P. Korfiatis *et al.*, "MRI texture features as biomarkers to predict MGMT methylation status in glioblastomas," *Medical physics,* vol. 43, no. 6Part1, pp. 2835-2844, 2016.

[2] L. Han and M. R. Kamdar, "MRI to MGMT: predicting methylation status in glioblastoma patients using convolutional recurrent neural networks," in *Pacific Symposium on Biocomputing 2018: Proceedings of the Pacific Symposium*, 2018: World Scientific, pp. 331-342.

[3] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, "Brain tumor classification using dense efficient-net," *Axioms,* vol. 11, no. 1, p. 34, 2022.

[4] M. K. Abd-Ellah, A. I. Awad, A. A. Khalaf, and H. F. Hamed, "A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned," *Magnetic resonance imaging,* vol. 61, pp. 300-318, 2019.

[5] G. Raut, A. Raut, J. Bhagade, J. Bhagade, and S. Gavhane, "Deep learning approach for brain tumor detection and segmentation," in *2020 International Conference on Convergence to Digital World-Quo Vadis (ICCDW)*, 2020: IEEE, pp. 1-5.

[6] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017: Ieee, pp. 1-6.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[8] P. Korfiatis, T. L. Kline, D. H. Lachance, I. F. Parney, J. C. Buckner, and B. J. Erickson, "Residual deep convolutional neural network predicts MGMT methylation status," *Journal of digital imaging,* vol. 30, pp. 622-628, 2017.

[9] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 11, pp. 2298-2304, 2016.

[10] Y. Huang, C. Du, Z. Xue, X. Chen, H. Zhao, and L. Huang, "What makes multi-modal learning better than single (provably)," *Advances in Neural Information Processing Systems,* vol. 34, pp. 10944-10956, 2021.

[11] A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, 2019: Springer, pp. 311-320.

[12] K.-L. Tseng, Y.-L. Lin, W. Hsu, and C.-Y. Huang, "Joint sequence learning and cross-modality convolution for 3D biomedical segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 6393-6400.

[13] R. Qu and Z. Xiao, "An attentive multi-modal cnn for brain tumor radiogenomic classification," *Information,* vol. 13, no. 3, p. 124, 2022.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[15] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[16] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," *Advances in neural information processing systems,* vol. 27, 2014.

[17] U. Baid *et al.*, "The RSNA-ASNR-MICCAI BraTS 2021 benchmark on brain tumor segmentation and radiogenomic classification," *arXiv preprint arXiv:2107.02314,* 2021.