Predicting Isocitrate Dehydrogenase 1 (IDH1) Mutation in Patients with Gliomas using a Novel Deep Learning Network

by

Haonan Xiao

Medical Physics Program
Duke Kunshan and Duke University

Date: _____

Approved:

_____
Zheng (Jim) Chang, Supervisor

_____
James Bowsher

_____
Lei Ren

Thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science in the Department of
Duke Kunshan and Duke University
Medical Physics Program

2019

ABSTRACT

Predicting Isocitrate Dehydrogenase 1 (IDH1) Mutation in Patients with Gliomas using a Novel Deep Learning Network

by

Haonan Xiao

Medical Physics Program
Duke Kunshan and Duke University

Date: _____
Approved:

_____
Zheng (Jim) Chang, Supervisor

_____
James Bowsher

_____
Lei Ren

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Duke Kunshan and Duke University Medical Physics Program

2019

Abstract

IDH is a gene that heavily affects the treatment response of gliomas and is associated with patient's prognosis. Convolutional neural networks (CNNs) showed great potential in predicting IDH mutations. However, these CNN models require time-intensive image-preprocessing before being used for predictions. There are two main purpose of this study. The first purpose is to investigate the feasibility of applying a novel convolutional neural network based on the Inception-ResNet to reduce image preprocessing steps and improve accuracy in IDH mutation prediction. The second purpose is to evaluate different data augmentation methods on brain studies.

MR images of 103 patients were selected from The Cancer Imaging Archive (TCIA). Intensity normalization of every individual slice is the only image pre-processing step. The T1w post-contrast, FLAIR, and T2w images at the same slice location were grouped together and considered as one training sample. This give rise to 209 IDH-mutant samples from 42 patients and 356 IDH-wild-type samples from 61 patients that were randomly selected to become training, validation, and test sets. To avoid overfitting in the model performance, data augmentation methods were applied individually to both training and validation sets in each training. The augmentation methods included duplication, noise addition, rotation, translation, cropping and mirroring. Images from one sample were fed to different input channels of Inception-

ResNet, and the predictions were based on the extracted features and the patient's age at diagnosis. Prediction accuracy was used to assess the performance of different augmentation methods.

With only intensity normalization, the proposed model using training sets augmented by rotation and noise addition achieved the IDH prediction accuracies of 91.8% and 91.8%, respectively. On the same training, validation and test sets, the proposed model trained on data augmented by duplication, cropping, translation, and mirroring gave accuracies of 81.6%, 79.6%, 83.7%, and 85.7%, respectively.

This work investigated the feasibility of the application of the novel convolutional neural network based on the Inception-ResNet on IDH mutation prediction, and high accuracies can be achieved with only intensity normalization as image preprocessing. Among all data augmentation methods, noise addition and rotation shows better performance and might suggest potential value for other clinical applications using machine learning algorithms.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to thank my thesis advisor Dr. Zheng Chang of the department of radiation oncology at Duke University. The direction of this thesis research was inspired by him, and he offered me continuous support during the research.

Secondly, I would like to thank the medical physics graduate program at Duke Kunshan University. The program gave me the opportunity to study abroad and provided me plenty of equipment budget for hardware support.

My sincere thanks also go to my classmates, for the research and paper discussions, for the encouragement when facing problems, and for all the fun we had together.

1. Introduction

1.1 The history of convolutional neural network

CNNs are deep learning networks and have shown great potential in image recognition. Instead of taking the whole image as input, like what artificial neural network (ANN) does, CNNs try to minimize the computational cost by using shift invariant perceptrons in multiple layers.[1] Their shared and translation invariant weights make CNNs also known as shift invariant or space invariant artificial neural networks (SIANN).[2]

The idea of perceptrons could be traced back to the brain studies in 1950s and 1960s. Hubel and Wiesel identified two types of visual cells in brains: simple cells (S-cells) and complex cells (C-cells).[3] S-cells extract local features, while C-cells tolerant the features' deformation, such as shifting and rotation. Every neuron takes the visual stimuli within a certain range, which is known as receptive field. The receptive field of neighboring neurons are similar and overlapping, and the systematical combination of the size and location of receptive fields forms a complete visual space.

The first convolutional neural network is the time delay neural network (TDNN) proposed in 1987, which was used to recognize phonemes in speech signals.[4] It utilized shared weights along the temporal dimension, which made the speech processing time-invariant. This was the early prototype of the shift-invariant filters in image recognition. However, TDNN was a feedforward network and all the weights had to be updated in a separate back-propagation process.

In 1989, Yann LeCun et all proposed a CNN to recognize hand-written ZIP code numbers and made all the weight automatically trainable via back-propagation.[5] Later, they further improved the network to a 7-layer CNN and named it as LeNet-5,[6] which contains 2 convolutional layers, 2 subsampling layers, 2 full connection layers, and 1 output layer. LeNet-5 had shown great performance on the recognition of hand-written numbers and was adopted by several banks to deposit checks. However, the input layer of LeNet-5 could only receive 32 by 32 black-and-white images. The processing of higher resolution and colorful images requires larger and more convolutional layers, so this network was constrained by computational cost.

The development of processors, especially graphic processing units (GPUs), boosted the development of CNNs. In 2004, K. S. Oh and K. Jung first implemented neural network training with GPUs instead of CPUs and got a 20 times faster performance.[7] Later, many other researchers followed this idea and implemented CNN training with GPUs.[8-11] In 2012, Alex Krizhevsky et al proposed a CNN model, AlexNet, and achieved a top-5 error rate of 15.3% in image recognition.[12] AlexNet was considered a classical CNN model and the majority of recent CNN models are fine-tuning or modification of AlexNet. By the end of 2018, the work has been cited more than 34,000 times.

1.2 The Structure of a Typical CNN: AlexNet

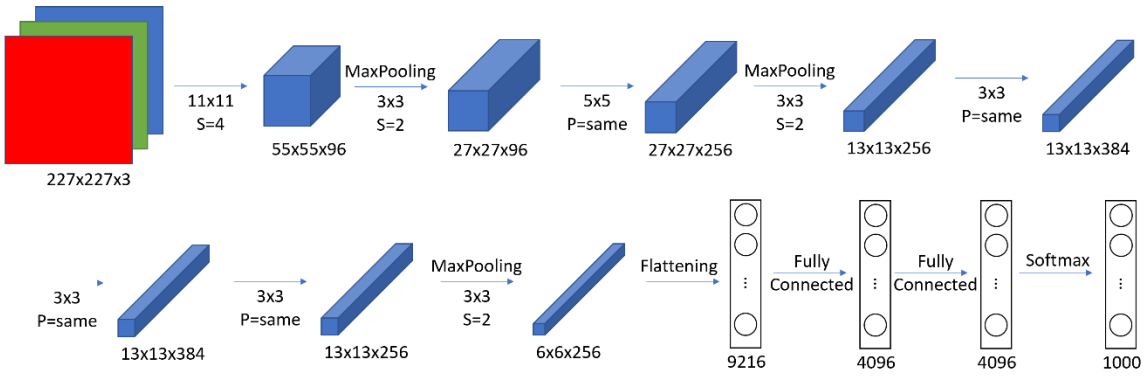

Figure 1: The Architecture of AlexNet.

Figure (1) shows the architecture of AlexNet. There are 5 different layers in AlexNet: input layer, convolutional layer, activation layer, pooling layer, fully connected layer, and output layer. These layers are widely utilized in different CNN models, and their function will be further explained in this session.

1.2.1 Input layer

This layer receives the input images. The input layer can be 2 dimensional and receive gray level images. If the input images are RGB images, the input layer should have 3 channels on the third dimension and receive the red, green, and blue channel of the RGB images, respectively. Usually, CNN models include a single input layer; however, if a few images are fed to the CNN model simultaneously, there can be multiple input layers.

1.2.2 Convolutional layer

Convolutional layer is the core of CNN models and Figure (2) demonstrates how it functions. The convolutional layer contains multiple kernels (also called filters, detectors). Each kernel K scans through the input volume I and multiply pixel value of I with the corresponding weight in K. The sum of all the multiplication products becomes one cell of the output. When the kernel completes scanning, an output volume is created, and the size of that volume is

$$S_{\text{output}} = S_I - S_K + 1 \quad (1)$$

where S_I and S_K are the size of input volume and kernel, respectively.

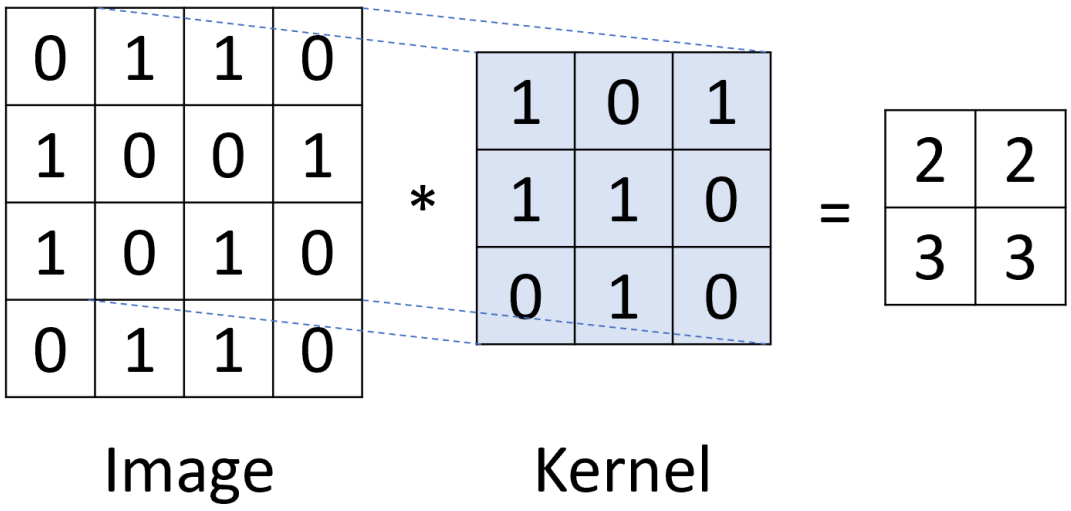

Figure 2: The demonstration of convolution.

The kernel can move through the input volume one pixel by one pixel, and it can also move several pixels at each step to avoid excess overlapping, especially when the

size of the kernel is large. The number of pixels each step goes through is call stride, and the size of the output volume now becomes

$$S_{output} = \frac{S_I - S_K}{s} + 1 \quad (2)$$

where s is stride.

As shown in Figure (2) and Eq (1)(2), the size of output volume is smaller than the input. After a few convolutions, the output volume may significantly shrink and cannot afford any further processing. Also, it makes CNN models over-sample the central area but under-sample the periphery. Padding is a popular choice to enlarge the size of output volume. As shown in Figure (3), 2 circles of zeros are attached to the original image and enlarge it from 32x32 to 36x36. With a padding of p circles, the size of output volume now becomes

$$S_{output} = \frac{S_I - S_K + 2p}{s} + 1 \quad (3)$$

In many CNN models, the padding value is not a specific number, but a general requirement called “same”, which keeps the length and width of output volume the same as those of the input volume.

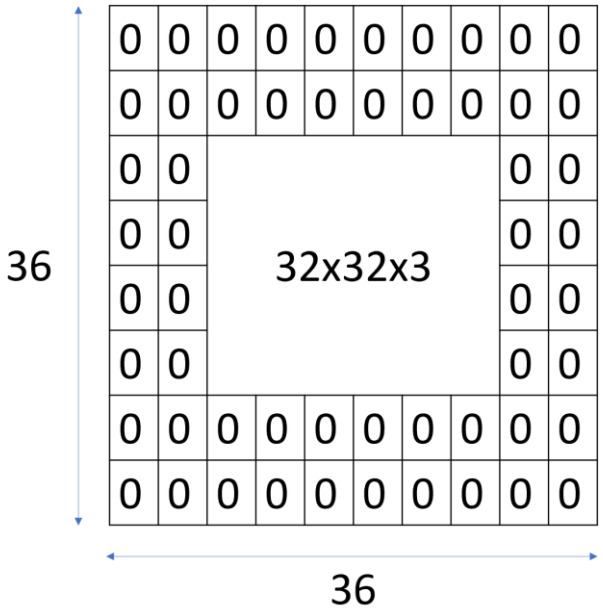

Figure 3: Padding makes a 32x32 image to be 36x36.

Eq (3) describes the first and second dimension of output volume, and the third dimension depends on the number of kernels. Each kernel forms a plane when it completes the scanning on the input volume, the planes from multiple kernels concatenate together to form a volume.

1.2.3 Activation layer

Neuronal networks require non-linearity, otherwise all the layers between the input and the output can be equivalent to a single matrix and therefore noneffective. For ANNs, the activation is usually sigmoid or tanh function, as shown in Eq (4) and Figure (4).

$$\begin{cases} \text{Sigmoid(x)} = \frac{1}{1+e^{-x}} \\ \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e_{-x}} \end{cases} \tag{4}$$

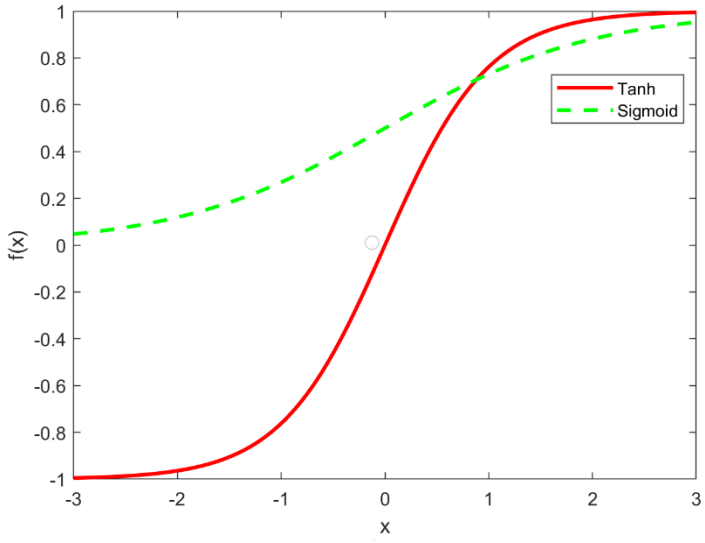

Figure 4: Sigmoid and tanh function.

The primary problem of these two activation functions is that gradients are too small when the variables are either small or large, making the training slow and difficult to converge. In 2011, Xavier Glorot et al proposed a new activation function, rectified linear unit (ReLU) function, and showed that ReLU function allows a faster and better training.[13] The mathematical expression of ReLU function is Eq (5) and Figure (5) shows the diagram.

$$\text{ReLU(x)} = \max(x, 0) \tag{5}$$

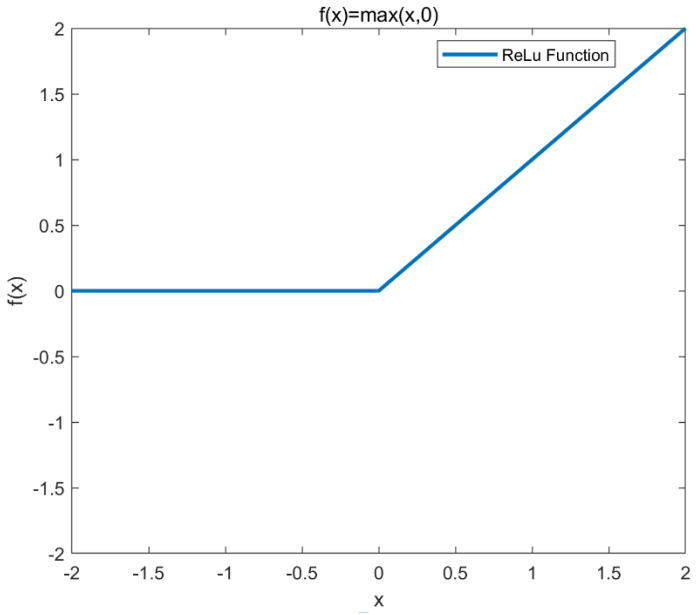

Figure 5: ReLU function.

After normalization, the output volume will be fed to an activation layer with ReLU function. Voxel with negative values are rectified to be 0, and voxels with positive values remain. Non-linearity is therefore introduced in CNN models.

1.2.4 Pooling layer

Pooling is another important component of CNNs, which introduce non-linearity by non-linear down-sampling. Besides, it also reduces the parameters and the computation in CNNs. Figure (6) shows a max pooling layer with filter size of 2 and stride of 2. For this max pooling layer, the filter extracts the maximal value within a 2x2 region on the input volume and moves 2 pixels horizontally or vertically at each step. Average pooling layer is also common in CNNs, whose filter extracts the mean value of a certain region. However, pooling layers only down-sample 2 dimensions: width and

height. Consequently, the size of the output volume after a pooling layer with filter size of F and stride of S changes to

$$\begin{cases} W_2 = \frac{W_1 - F}{S} + 1 \\ H_2 = \frac{H_1 - F}{S} + 1 \\ D_2 = D_1 \end{cases} \tag{6}$$

where W, H, and D represent the width, height, and depth of the matrix and the subscription 1 and 2 represent the input and output volume, respectively.

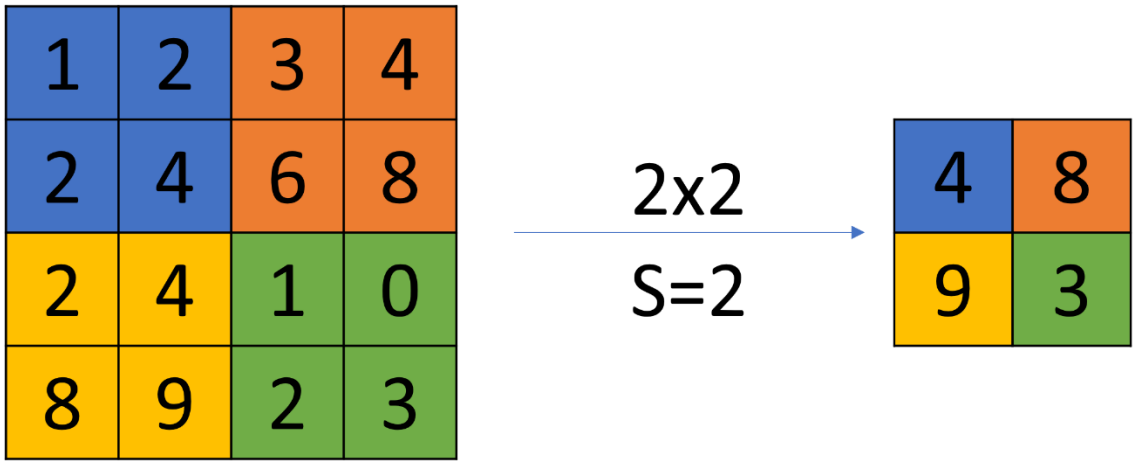

Figure 6: Max pooling layer, with a filter size of 2 and a stride of 2

1.2.5 Fully connected layer

After several convolutional and pooling layers, the extracted features can be combined with fully connected layers, which are common structures in ANN. Before fed to the fully connected layers, the extracted feature volume has to be flattened and become one-dimensional arrays. In AlexNet, there are three fully connected layers, including the

output layer. However, recent CNN architectures, such as ResNet[14], tend to replace fully connected layer with a global average pooling layer, since fully connected layers are very computationally expensive.

1.3 Advanced CNN architectures

AlexNet was acknowledged as a classical CNN architecture, and a great majority of recent CNN models were based on it. They achieved better recognition performance by adjusting filters and adding more convolutional layers, such as VGG-16 and VGG-19.[15] Some other CNN models, however, modified the fundamental structure of AlexNet and achieved even better results. ResNet and GoogleNet were typical representations of these CNN models.

1.3.1 Residual Networks

ResNet was proposed because more layers in AlexNet-like networks, or "plain" networks, led to poorer performance when the number of layers reached a certain limit.[14] The primary cause is the vanishing gradient problem. This problem was first identified in 1991[16]: in neural network training with gradient-based learning methods and backpropagation, the gradient can be vanishingly small, especially when the computation of gradient goes through many layers and thus many times of the chain rule. Because of ReLU activation function, the gradient is large enough for 20-layer networks, but it cannot afford any deeper layers.

ResNet keeps the gradient by adding shortcuts and bypassing some convolutional layers, as shown in Figure (7). The shortcuts are actually identity functions, which

directly add the input volume to several layers deeper without any processing. When computing gradient, it will be multiplied by the derivative of identity functions, which is simply 1. Therefore, the gradient is large enough for training even for 152-layer networks.[14]

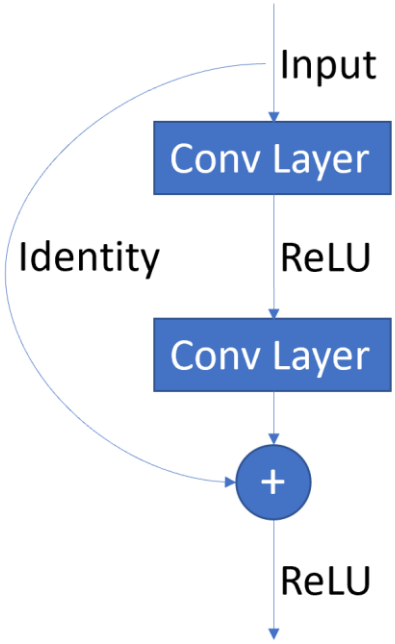

Figure 7: The demonstration of residual learning.

1.3.2 Inception module

Inception module is the primary breakthrough of GoogleNet.[17] In CNN models, the size and number of convolutional filters significantly influence the performance, and it takes the researchers many tries to find out the best solution. GoogleNet solved this problem by utilizing inception modules, as shown in Figure (8). Instead of

consequentially going through each layer, input volume is processed simultaneously in parallel branches, and the size and numbers of convolutional filters in each branch can be different. During training, the weight of each branch will be updated automatically, and the best solution can be obtained.

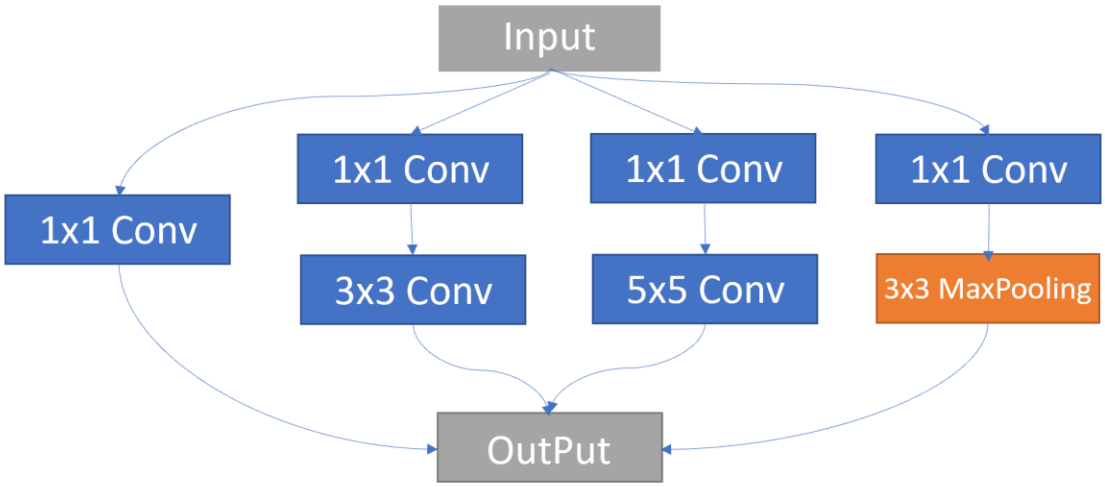

Figure 8: Demonstration of inception module in GoogleNet.

1.3.3 Inception-ResNet

In 2017, Christian Szegedy et al combined ResNet and inception module and proposed Inception-ResNet. The architecture of Inception-ResNet is shown in Figure (9). Inception-ResNet adds shortcuts as an additional branch, and therefore it has the properties both from ResNet and inception module, which helped it achieved improved performance in image recognition.

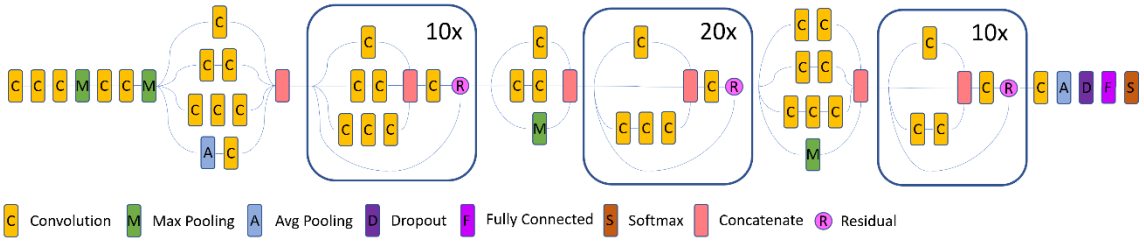

Figure 9: The architecture of Inception-ResNet.

1.4 Data augmentation

The idea of data augmentation can be traced back to 2003, when Patrice Y. Simard et al were training CNN for digit recognition and faced the problem of limited data.[24] They generated various elastic deformation maps and applied them to original images. Neural networks consider the deformed images as new images, since the pixel values received by filters are different. In this way, they successfully enlarged their training data set and achieved record-breaking accuracy.[18]

Recent data augmentation methods are more diverse, including cropping, mirroring, translation, rotation of the original image and noise addition. For RGB images, color shifting, and gray scale are also practical, as shown in Figure (10). The purpose of data augmentation is to make the network insensitive to a certain change in the input images. For example, if the input images are augmented by rotation, the network will manage to predict consistent result even if the input images are rotated. Consequently, the augmentation can make the prediction rotation-invariant. By augmented the training images with different methods, the network can be more robust and make reliable predictions in reality.

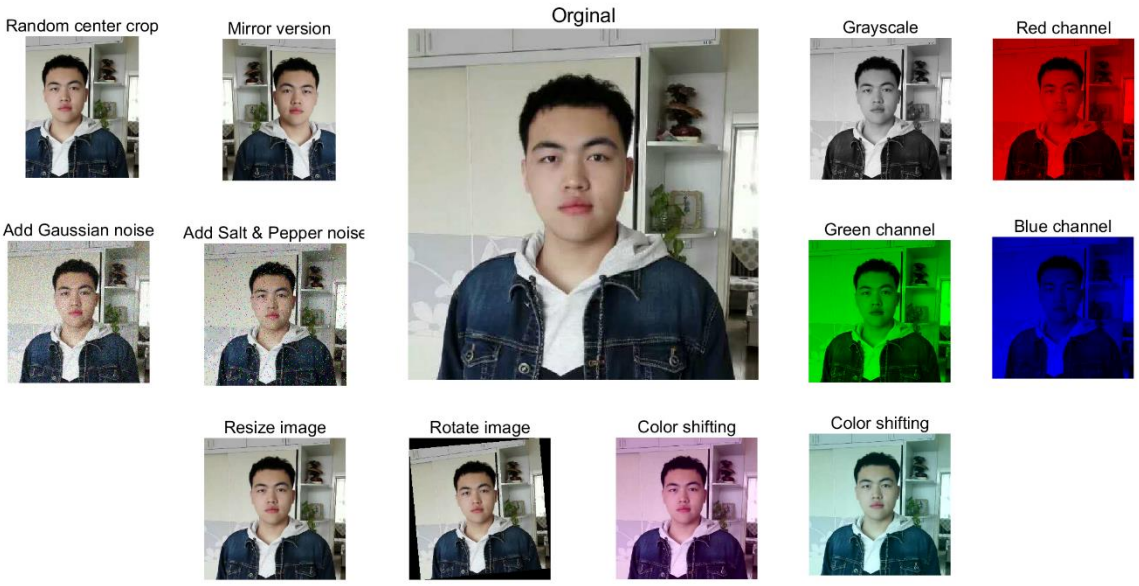

Figure 10: Examples of recent data augmentation methods for image recognition.

1.5 Optimizer

With a sample fed to the neural network, an objective function (also called loss function or error function) is calculated based on the different between the network prediction and the ground truth. The gradient, or the partial derivations about neural network parameters, can be calculated. Neural networks then update their weights by gradient-based backpropagation methods, which are also known as optimizers. Gradient descent (GD) is a popular optimizer in many fields, and it has many extensions or variants in neural networks.

1.5.1 Stochastic gradient descent (SGD)

For the conventional GD, it minimizes the average of the loss function of all training samples, i.e.,

$$\mathrm{Q(w)} = \frac{1}{\mathrm{n}}\sum_{i=1}^{n} Q_i(w) \tag{7}$$

where $\mathrm{Q_i}$ is the loss function of the ith sample, and n is the number of samples. In order to minimize $\mathrm{Q(w)}$, the parameter w should be updated as

$$\mathrm{w} \coloneqq \mathrm{w} - \eta\nabla\mathrm{Q(w)} = \mathrm{w} - \eta\sum_{i=1}^{n}\frac{\nabla Q_i(w)}{n} \tag{8}$$

where η is the step size (usually called the learning rate in machine learning).

However, computing the sum of the gradient over all training samples can be very computationally expensive and sometimes impossible. In 1951, Herbert Robbins et al proposed a new method,[19] which does not average on all training samples but only compute the gradient from one sample, i.e.,

$$\mathrm{w} \coloneqq \mathrm{w} - \eta\nabla\mathrm{Q_i}(w) \tag{9}$$

where $\mathrm{Q_i}(w)$ is the loss function of the ith sample. For an SGD optimization, the training samples will be randomly shuffled, therefore, Eq (9) is a stochastic process. However, the gradient from a single training sample fluctuates and can mislead the training. A compromised strategy is to compute the gradient from a subset of all training samples, which is known as a mini-batch.[20]

1.5.2 SGD with momentum

Later, Herbert Robbins et al proposed a new SGD backpropagation method with momentum to further stabilize the gradient. In this method, the previous parameter update Δw is kept to the next iteration, i.e.,

$$\begin{cases} \Delta w \coloneqq \alpha \Delta w - \eta \nabla Q_i(w) \\ w \coloneqq w + \Delta w \end{cases} \tag{10}$$

where α is the momentum parameter. It gets the name "momentum" because the term αΔw gives the weights an inertia and it will keep the moving direction until the loss function gradient, or the "force", changes it.[19]

1.5.3 Adam

Adam is the abbreviation of Adaptive Moment Estimation. It was proposed by DP kingma et al in 2014.[21] In Adam, the learning rate is adapted for each of the parameters and therefore, the training can be more specific and effective. For the t th iteration in Adam, weight $w^{(t)}$ gives the loss function $L^{(t)}$, and the weight in the new iteration $w^{(t+1)}$ can be updated as

$$\begin{cases} m_w^{(t+1)} = \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \\ v_w^{(t+1)} = \beta_2 v_w^{(t)} + (1-\beta_2)\left(\nabla_w L^{(t)}\right)^2 \\ \hat{m}_w = \frac{m_w^{(t+1)}}{1-(\beta_1)^{t+1}} \\ \hat{v}_w = \frac{v_w^{(t+1)}}{1-(\beta_2)^{t+1}} \\ w^{(t+1)} = w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w}+\epsilon} \end{cases} \tag{11}$$

where ϵ is a small parameter to avoid division by 0, and β_1 and β_2 are the forgetting factors for gradients and second moments of gradients, respectively. Adam has

been the most widely applied optimizer in neural network training for many years and researchers have cited the original paper more than 17,000 times.

1.6 CNN applications in gliomas classification

1.6.1 Gliomas

Gliomas are brain tumors that start in the glial cells. They account for 80% of malignant brain tumors.[22] Some studies show that the treatment response of gliomas associates with the isocitrate dehydrogenase 1 (IDH1) enzyme.[23-25] Mutation can occur at a single amino acid residue of the gene that encodes the enzyme, and the resulting mutant enzyme loses the ability to catalyze conversion of isocitrate to α-ketoglutarate. However, the mutation allows the enzyme to catalyze the production of D-2-hydroxyglutarate (D-2HG). Molenaar et al suggest that the accumulated D-2HG suppresses DNA repair and causes increased DNA damages in cancer cells after irradiation or chemotherapy.[24] Therefore, patients with mutant IDH1 are generally more sensitive to radiotherapy or chemotherapy and survive longer. Figure (11) visualizes the whole process.

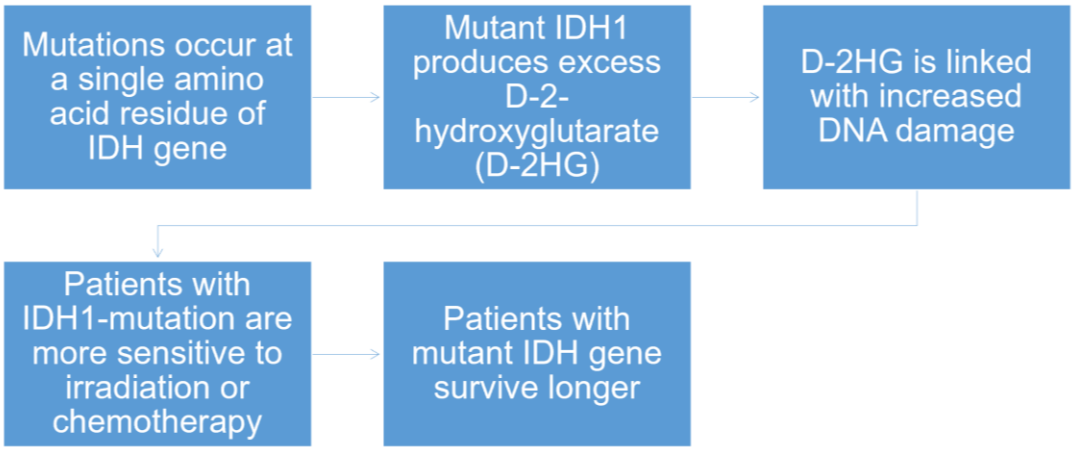

Figure 11: How IDH mutation affect patient's survival.

Considering the significant prognosis difference caused by IDH1, in 2016, the World Health Organization (WHO) divided grades II, III, and IV gliomas into two subgroups, which are IDH mutant and IDH wild-type gliomas.[26] The "gold standard" to determine IDH mutations are immunohistochemistry and genomic sequence analysis.[27,28] However, these methods require histopathological specimens and are therefore invasive and time-intensive. In addition, these researchers report that intratumoural genetic heterogeneity can make the biopsy-based genetic analysis incorrect, since it only samples a small part of the tumors.

1.6.2 CNN models to predict IDH 1 mutations

Medical imaging procedures are promising alternatives for IDH detection because they are non-invasive, repeatable and examine the whole tumor at once. Magnetic resonance imaging (MRI) is the most common diagnostic imaging modality for gliomas in routine clinical practice, and IDH mutant and IDH wild-type gliomas show different

appearance on MRI images. For example, (1) the involvement of frontal lobe is predominant in IDH mutant gliomas;[29] (2) T2 weighted imaging (T2WI) and fluid-attenuated inversion recovery (FLAIR) mismatching occurs primarily in IDH mutant gliomas;[30] (3) IDH mutant gliomas often show well-defined borders;[37] (4) the IDH mutation follows an age distribution and IDH mutant gliomas are more likely to be diagnosed in patients who are younger.[29] These pathological properties of IDH mutant and wild-type gliomas provide the possibility of non-invasive predictions.

Deep learning, especially convolutional neural network (CNN), has shown great potential in medical image recognition.[31-33] Some recent studies utilized ResNet on IDH mutation prediction and get high accuracy.[34-36] However, these prediction models require complicated image pre-processing, such as isotropic resampling, whole-brain extraction, image registration, and segmentation. These pre-processing procedures, especially the segmentation, can take up to half an hour and limit the clinical application.[37]

The goal of this thesis research is to investigate the feasibility of the new network architecture to reduce image preprocessing and improve classification accuracy on the IDH mutation prediction, and to evaluate the effectiveness of different augmentation methods in brain studies.

2. Methods and Materials

2.1 Patients Data

The MRI images of glioma patients were provided by The Cancer Imaging Archive (TCIA)[38], and their genetic information was provided by The Cancer Genome Atlas (TCGA). The research involves patients who met the following criteria: (1) IDH mutation or wild type were documented. (2) Image series included T1w-post-contrast, FLAIR, and T2w imaging. (3) Slice locations were provided in the DICOM files. (4) Tumor locations were known. (5) Patients ages were known. In total, 103 patients were selected. For every selected patient, his/her T1w-post-contrast, FLAIR, and T2w images of the same slice location were assembled as one training sample. 209 IDH-mutant samples from 42 patients and 356 IDH-wild-type samples from 61 patients were generated. With all the samples, a test set was selected, which contained 24 samples from 6 IDH-mutant patients and 25 samples from 6 IDH-wild-type patients, and the patients were randomly selected to avoid overlapping in the training and test sets.

2.2 Image Pre-processing

Intensity normalization was the only pre-processing applied in this study. For every individual MRI image, the intensities of pixels are normalized by subtracting the average value and divided by the standard deviation of them. The purpose of intensity normalization is to remove the background bias of MRI images.

2.3 Data Augmentation

Several data augmentation methods were involved in this research, including (Figure (1)): (1) Rotation, which randomly rotated the brain images from 0 to 10 degrees, counterclockwise; (2) Translation, which translated the brain images within 10 pixels horizontally and vertically, independent from each other; (3) Cropping, which cropped fractions of 200x200 pixels from the 256x256 original images. This method is also known as zooming in other literatures; (4) Noise addition, which scaled the background noise acquired from a phantom MRI scan (on a clinical 3T Siemens Skyra MR scanner) to different intensity levels and added it to the original images; (5) Mirroring, which flipped the brain images about the vertical axis. Mirroring was applied together with duplication to reach the required augmentation. Specifically, in order to augment the image 20 times, the brain image was first flipped, and the original and flipped images were both duplicated 10 times. (6) Duplication, which identically copied the brain images as a control group for comparison. Since the number of training samples were not equal in the two groups, IDH-mutant images are augmented 20 times while IDH-wild-type images were augmented 12 times to balance the quantity of the samples. In addition, all augmentation methods were combined with a ratio of 20% each to investigate the performance of mixed augmentation methods.

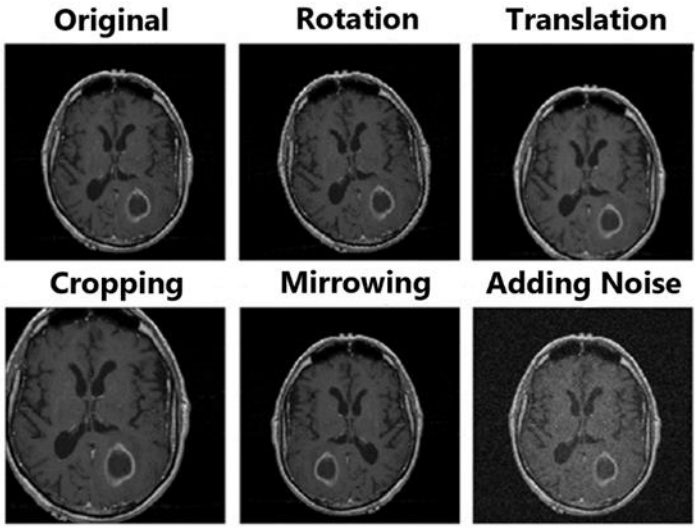

Figure 12: Demonstration of augmented images.

2.4 Deep Learning Model Based on Inception-ResNet

Inception-ResNet combines the architecture of Inception module and Residual CNN, as shown in Figure (9). Inception module is the parallel branches, in which different convolutional filters are applied. During training, the weight of each branch was updated in every iteration and finally the model could find the best branch, i.e., the best filters, and gave them the highest weight. The Residual CNN was demonstrated with the shortcuts ending with purple circles in Figure (9), which bypass several convolutional layers. The bypasses could conduct the gradient more efficiently in deep networks and make the training converge faster. Inception-ResNet has the advantages of both inception module and residual networks. It can automatically select the best

convolutional filters during the training and converge fast even with deep network layers.

The model comprised 3 parallel channels, which received T1 post-contrast, T2, and FLAIR images from one training sample, respectively. The extracted features were concatenated at the fully connected layer together with patient's age at diagnosis, on which the predictions were based (Figure (13)).

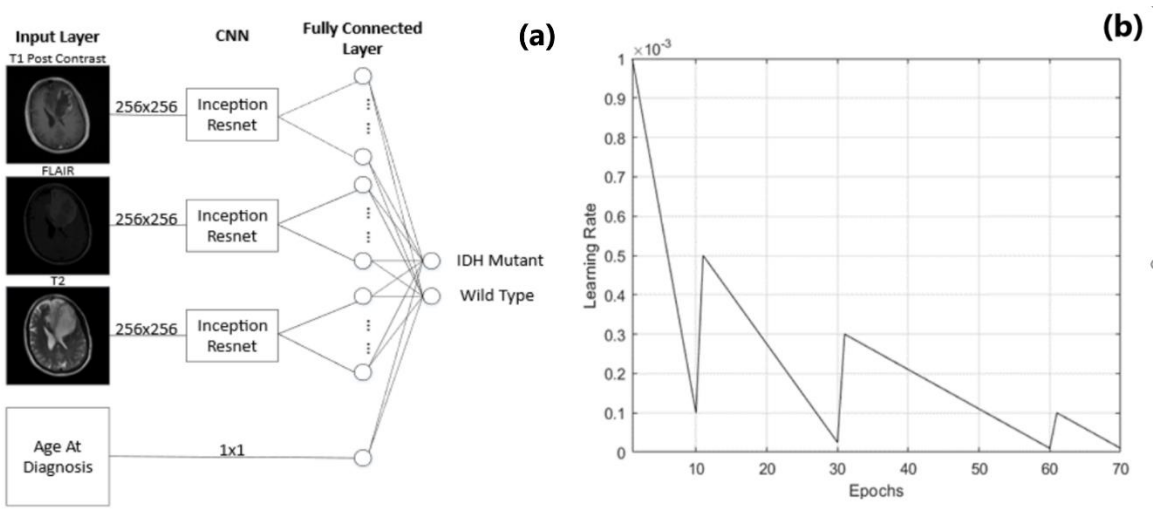

Figure 13: (a) The Inception-ResNet based deep learning model. (b) The learning rate schedule.

To evaluate the contribution of each channel, i.e., each imaging modality, one of the 3 parallel channels was taken out correspondingly in 3 experiments. More specifically, with noise addition-based augmentation, the combinations of T1 post-contrast and FLAIR, T1 post-contrast and T2, and FLAIR and T2 imaging were examined

in the 3 individual experiments. After the most contributing imaging modality was determined, it was separately evaluated.

2.5 Implementation Details

Learning rate is an important hyper-parameter in deep learning because it heavily influences the convergence of the objective function. The network training can take excess time or miss the global minimum if the learning rate is too small or too big. A common strategy is to monotonically decrease the learning rate during training, however, the training may be stuck in different local minimums or saddle points since the initial normalizations are random. and gives non-reproducible results.[39] Leslie N. Smith et al proposed a new strategy called cyclical learning rates, in which the learning rate is scheduled to decrease and increase in several periods.[40] As shown in Figure (13b), the learning rate overall decreases with several sharp increasing stages, so that the optimizer can escape from local minimums and finally converge to the global minimum.

To evaluate its robustness, the proposed the deep learning model was tested again with another pair of randomly selected training and test data sets sampled within the 103 patients described above. In the second test, rotation or noise addition was used to augment the training data sets.

In the training, the Adam algorithm optimized the binary cross-entropy objective function with a mini-batch size of 8. The training ended when it reached the 70th epoch or early stopped once the training loss reaches 10^{-5}. The training was based on Keras

(version 2.2.2) with TensorFlow (version 1.10.0) as backend and was conducted in Spyder IPython IDE (version 3.0.0). The network was trained on a GTX 1080 GPU card in Windows 10 (version 1803).

2.6 Evaluation of The Deep Learning Model

The performance the proposed deep learning model was evaluated by the prediction accuracy, sensitivity (True Positive Fraction, TPF) and specificity (True Negative Fraction, TNF) on the test set. In addition, receiver operating characteristic (ROC) curves were generated for each test, and the area under curve (AUC) was another parameter for evaluation. Among all evaluation factors, prediction accuracy is of priority.

3. Results

Every training epoch typically took 900 seconds, and the test set took about 60 seconds. The performance of the proposed Inception-RessNet based model using different augmentation methods was shown in Table (1). The prediction accuracies of duplication, cropping, combination, translation, mirroring, rotation, noise addition were 81.6%, 79.6%, 81.6%, 83.7%, 85.7%, 91.8%, and 91.8%, respectively. Compared to duplication, in which no actual new information is introduced, all data augmentation methods, except cropping, showed improved performance. The TPF showed the most significant improvement. Among all the methods, random rotation and noise addition gave the best accuracies.

Table 1: The performance of different augmentation methods

Augmentation Methods	TPF	TNF	Accuracy	AUC
Duplication	62.5%	100%	81.6%	0.95
Cropping	70.8%	88.0%	79.6%	0.91
Combination	75.0%	88.0%	81.6%	0.91
Translation	79.2%	88.0%	83.7%	0.94
Mirroring	70.8%	100%	85.7%	0.99
Rotation	83.3%	100%	91.8%	1.00
Noise Addition	83.3%	100%	91.8%	0.99

To evaluate the robustness of the proposed the deep learning model, it is tested again with another pair of randomly selected training and test data sets. The results of the two tests were summarized in Table (2). As shown, the performance of the proposed deep learning model is generally consistent in the two tests. More specifically, high and consistent prediction accuracies were achieved, 91.8% and 90.2% from rotation-based augmentation and 91.8% and 88.2% from noise addition-based augmentation, respectively. The specificity, also known as TNF, was much higher than sensitivity (TPF). Figure (14) showed the receiver operation characteristic (ROC) curves of the models (a) trained on training set 1 augmented by rotation (b) trained on training set 2 augmented by rotation; (c) trained on training set 1 augmented by noise addition;(d) trained on training set 2 augmented by noise addition. The AUC (area under curve) values were 1.00 and 0.95 from rotation and 0.99 and 0.95 from noise addition, respectively.

Table 2: The test results on the robustness of the proposed deep learning model

Test	Augmentation	TPF	TNF	Accuracy	AUC
1	Rotation	83.3%	100.0%	91.8%	1.00
	Noise Addition	83.3%	100.0%	91.8%	0.99
2	Rotation	82.1%	100.0%	90.2%	0.99
	Noise Addition	78.6%	100.0%	88.2%	0.95

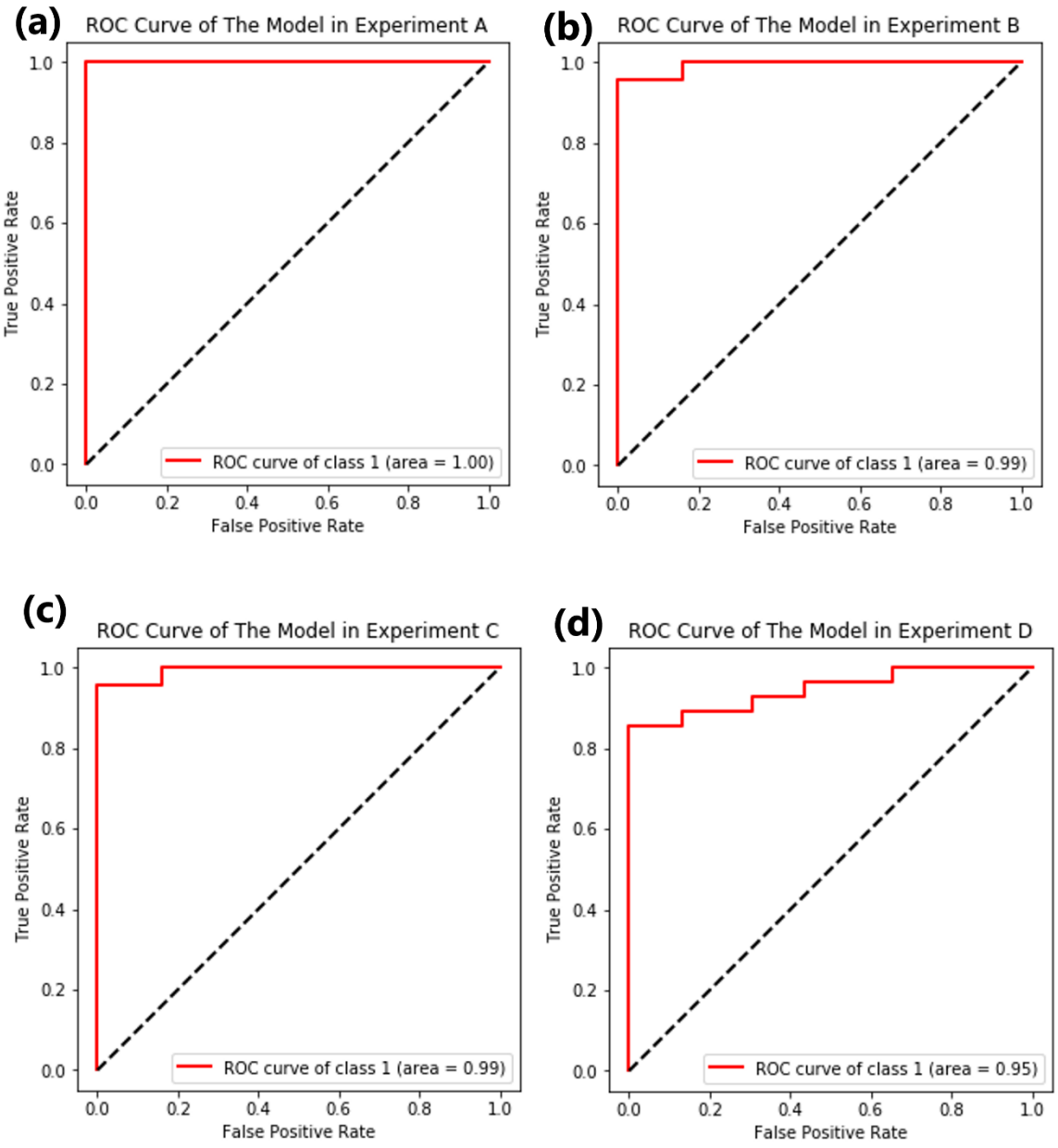

Figure 14: The ROC curves of the models.

The prediction performance with one channel taken out was shown in Table (3). The two tests that included T1 post-contrast imaging gave high accuracies and they were comparable with the results in Table (1), despite the AUC values were slightly lower. However, when T1 post-contrast imaging was taken out, as shown in the third test, the prediction accuracy and AUC value fell to 79.6% and 0.80, respectively.

Table 3: The prediction performance with one channel taken out. The training samples were augmented by noise addition.

Test	Remaining modalities	TPF	TNF	Accuracy	AUC
1	T1 post-contrast, FLAIR	83.3%	100%	91.8%	0.98
2	T1 post-contrast, T2	79.2%	100%	89.8%	0.94
3	FLAIR, T2	75.0%	84.0%	79.6%	0.80

After these 3 experiments, another test was carried out in which T1 post-contrast was the only involved imaging modality. The result was shown in Table (4).

Table 4: The prediction performance with the most contributing channel. The training samples were augmented by noise addition.

Test	Remaining modalities	TPF	TNF	Accuracy	AUC
4	T1 post-contrast	66.7%	100%	83.7%	0.99

4. Discussion

In this study, the utility of an Inception-ResNet based deep learning model was verified to minimize image pre-processing in the prediction of IDH mutations. To our best knowledge, it is the first study to investigate the feasibility of reducing image pre-processing steps while maintaining high prediction accuracy. The prediction accuracies of the proposed deep learning model were high and consistent with only intensity normalization as image pre-processing. It suggests that the new CNN architecture, i.e. Inception-ResNet, has the capacity to use minimally pre-processed images, which makes deep learning more practical in clinical applications since the data can be fed to the network more quickly and conveniently. The quick and accurate IDH mutation predictions may benefit clinical decision making because it guides patient management and treatment choice.

Various deep learning models were trained with different imaging modalities, namely the combination of T1 post-contrast and FLAIR, T1 post-contrast and T2, and FLAIR and T2 imaging. The results in Table (3) suggests that T1 post-contrast imaging contributes the most to the predication making. Chang *et al.* reported similar result.[34] This is probably because T1 post-contrast imaging offers the clearest tumor-normal tissue boundary and contrast among all three imaging modalities.

The TNF of the prediction results are always higher than the TPF, even reaching 100% at times. It is also noted that the AUC value sometimes achieves 1.00, which means there is a certain threshold that can completely separate IDH mutant and wild-type gliomas. Specifically, in the experiment showed in Figure (14a), the predictions can be 100% accurate when the threshold is 0.963, a very large number. It implies the models tend to predict negative results, i.e., IDH wild-type, unless they are very confident the prediction is positive. A potential explanation can be the natural possibility of the occurrence of IDH mutation. IDH wild-type are predominant in gliomas,[29] which is also true for the training samples in this study. Although the IDH wild-type and IDH mutant glioma images were augmented different numbers of times to balance the number of training samples, the IDH wild-type group does contain more information since it comprises more patients. Therefore, for a given sample, a negative prediction is more likely to be accurate and the deep learning model probably learned this tendency during training.

One common challenge in deep learning model training is the limitation of data and a well acknowledged strategy is data augmentation. The effectiveness of various augmentation methods was evaluated, including rotation, translation, cropping, mirroring, and noise addition. Data augmentation makes the model insensitive to a certain change in the input images. For example, input images augmented by translation make the prediction consistent even if the images are shifted. Random rotation and noise addition gave the best performance probably because these two scenarios are more common in the practice. The evaluation of different augmentation methods can benefit the brain deep learning studies in the future. As shown in Table (3), the involvement of non-effective augmentation methods leads to compromised performance. It suggests the deep learning projects can be more efficient if augmentation methods are evaluated before implementation.

This deep learning model can potentially aid the clinical diagnosis and prognosis prediction. IDH mutation has been an important factor that affects clinical decision making, for example, the presence of such mutations can be a sign of infiltrating gliomas, i.e., WHO grade > I.[41] Although immunohistochemistry and DNA sequence analysis can provide almost 100% reliable result, they are invasive procedures and take a lot of time and money. With the help of deep learning predictions, patient's genetic information can be obtained in a more convenient and efficient way. Also, tumors with IDH mutations are generally much less aggressive than those without mutations, and

these mutations are consider powerful prognosis biomarker.[41] Clinical staffs, such as physicians, may make more appropriate clinical decisions when this factor can be provided by the deep learning model instead of time-consuming biopsies.

Compared to those existing deep learning models, the biggest novelty of this Inception-ResNet based deep learning model is the capability of processing brain images without pre-processing. Those mentioned deep learning models in the introduction can achieve excellent IDH mutation prediction accuracies, however, the pre-processing, including isotropic resampling, field bias correction, image registration, skull-stripping, and segmentation, can be a barrier for the clinical applications. Among the preprocessing steps, segmentation is of special concern because it is the most labor-intensive one. With the capability of processing raw images, this deep learning model shows a better potential in clinical application.

There are several limits of this study. First, a larger population of the test subjects may further improve the performance of the models. The relatively few subjects might be the reason of the extreme AUC value in Figure (14a). Second, more channels can be added to the network architecture. In our study, there are only 3 channels for T1w-post-contrast, FLAIR and T2w axial images. Chang, K., et al. showed the combination of images in axial, coronal, and sagittal dimensions may improve the model performance.[34] TCIA provides T1, T2, FLAIR, perfusion and diffusion images in sagittal and coronal dimensions, however, not every patient has all those series. In order to keep the training

samples consistent, we did not utilize coronal and sagittal images. Third, the feasibility of the network can be further verified with more clinical data. Rotation and noise addition gave the best performance on brain studies, however, it may not be true for abdominal or breast studies.

5. Conclusion

In this study, I built a novel CNN model base on Inception-ResNet that can simultaneously receive T1 post-contrast, FLAIRE, and T2 images of glioma patients. This model verified that IDH status could be predicted at a high accuracy with minimal image pre-processing. Besides, among all data augmentation methods, it has been found that slight rotation and noise addition could be considered effective augmentation methods for brain studies.

References

1. Yann LJU. LeNet-5 convolutional neural networks. 2015.

2. Zhang W, Itoh K, Tanida J, Ichioka YJAo. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. 1990;29(32):4790-4797.

3. Hubel DH, Wiesel TNJTJop. Receptive fields and functional architecture of monkey striate cortex. 1968;195(1):215-243.

4. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ. Phoneme recognition using time-delay neural networks. In: *Readings in speech recognition.* Elsevier; 1990:393-404.

5. LeCun Y, Boser B, Denker JS, et al. Backpropagation applied to handwritten zip code recognition. 1989;1(4):541-551.

6. LeCun Y, Bottou L, Bengio Y, Haffner PJPotI. Gradient-based learning applied to document recognition. 1998;86(11):2278-2324.

7. Oh K-S, Jung KJPR. GPU implementation of neural networks. 2004;37(6):1311-1314.

8. Chellapilla K, Puri S, Simard P. High performance convolutional neural networks for document processing. Paper presented at: Tenth International Workshop on Frontiers in Handwriting Recognition2006.

9. Hinton GE, Osindero S, Teh Y-WJNc. A fast learning algorithm for deep belief nets. 2006;18(7):1527-1554.

10. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. Paper presented at: Advances in neural information processing systems2007.

11. Poultney C, Chopra S, Cun YL. Efficient learning of sparse representations with an energy-based model. Paper presented at: Advances in neural information processing systems2007.

12. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Paper presented at: Advances in neural information processing systems2012.

13. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. Paper presented at: Proceedings of the fourteenth international conference on artificial intelligence and statistics2011.

14. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Paper presented at: Proceedings of the IEEE conference on computer vision and pattern recognition2016.

15. Simonyan K, Zisserman AJapa. Very deep convolutional networks for large-scale image recognition. 2014.

16. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: A field guide to dynamical recurrent neural networks. IEEE Press; 2001.

17. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. Paper presented at: Proceedings of the IEEE conference on computer vision and pattern recognition2015.

18. Simard PY, Steinkraus D, Platt JC. Best practices for convolutional neural networks applied to visual document analysis. Paper presented at: null2003.

19. Robbins H, Monro S. A stochastic approximation method. In: *Herbert Robbins Selected Papers.* Springer; 1985:102-109.

20. Bottou L, Bousquet O. The tradeoffs of large scale learning. Paper presented at: Advances in neural information processing systems2008.

21. Kingma DP, Ba JJapa. Adam: A method for stochastic optimization. 2014.

22. Goodenberger ML, Jenkins RB. Genetics of adult glioma. *Cancer genetics.* 2012;205(12):613-621.

23. Dang L, White DW, Gross S, et al. Cancer-associated IDH1 mutations produce 2-hydroxyglutarate. *Nature.* 2009;462(7274):739.

24. Molenaar RJ, Maciejewski JP, Wilmink JW, van Noorden CJF. Wild-type and mutated IDH1/2 enzymes and therapy responses. *Oncogene.* 2018;37(15):1949-1960.

25. Yan H, Parsons DW, Jin G, et al. IDH1 and IDH2 mutations in gliomas. *New England Journal of Medicine.* 2009;360(8):765-773.

26. Louis DN, Perry A, Reifenberger G, et al. The 2016 World Health Organization classification of tumors of the central nervous system: a summary. 2016;131(6):803-820.

27. Kickingereder P, Sahm F, Radbruch A, et al. IDH mutation status is associated with a distinct hypoxia/angiogenesis transcriptome signature which is non-invasively predictable with rCBV imaging in human glioma. 2015;5:16238.

28. Suh CH, Kim HS, Jung SC, Choi CG, Kim SJ. Imaging prediction of isocitrate dehydrogenase (IDH) mutation in patients with glioma: a systemic review and meta-analysis. *European Radiology.* 2018.

29. Lai A, Kharbanda S, Pope WB, et al. Evidence for sequenced molecular evolution of IDH1 mutant glioblastoma from a distinct cell of origin. 2011;29(34):4482.

30. Patel SH, Poisson LM, Brat DJ, et al. T2-FLAIR mismatch, an imaging biomarker for IDH and 1p/19q status in lower grade gliomas: a TCGA/TCIA project. 2017:clincanres. 0560.2017.

31. Liu S, Zheng H, Feng Y, Li W. Prostate cancer diagnosis using deep learning with 3D multiparametric MRI. Paper presented at: Medical Imaging 2017: Computer-Aided Diagnosis2017.

32. Nakao T, Hanaoka S, Nomura Y, et al. Deep neural network-based computer-assisted detection of cerebral aneurysms in MR angiography. *J Magn Reson Imaging.* 2018;47(4):948-953.

33. Wang SH, Phillips P, Sui YX, Liu B, Yang M, Cheng H. Classification of Alzheimer's Disease Based on Eight-Layer Convolutional Neural Network with Leaky Rectified Linear Unit and Max Pooling. *J Med Syst.* 2018;42(5):11.

34. Chang K, Bai HX, Zhou H, et al. Residual Convolutional Neural Network for the Determination of IDH Status in Low- and High-Grade Gliomas from MR Imaging. *Clin Cancer Res.* 2018;24(5):1073-1081.

35. Chang P, Grinband J, Weinberg B, et al. Deep-Learning Convolutional Neural Networks Accurately Classify Genetic Mutations in Gliomas. *American Journal of Neuroradiology.* 2018.

36. Li Z, Wang Y, Yu J, Guo Y, Cao W. Deep learning based radiomics (DLR) and its usage in noninvasive IDH1 prediction for low grade glioma. *Scientific reports.* 2017;7(1):5467.

37. Lustberg T, van Soest J, Gooding M, et al. Clinical evaluation of atlas and deep learning based automatic contouring for lung cancer. 2018;126(2):312-317.

38. Clark K, Vendt B, Smith K, et al. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. 2013;26(6):1045-1057.

39. Dauphin YN, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. Paper presented at: Advances in neural information processing systems2014.

40. Smith LN. Cyclical learning rates for training neural networks. Paper presented at: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on2017.

41. Horbinski C. What do we know about IDH1/2 mutations so far, and how do we use it? *Acta neuropathologica.* 2013;125(5):621-636.